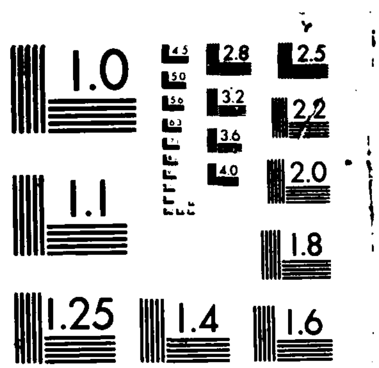
AD-A185 247 AUTOMATING RESPONSE SURFACE METHODOLOGY FOR THE ARSENAL EXCHANGE MODEL(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. G E YIELDING 1/4

UNCLASSIFIED MAY 87 AFIT/GOR/ENS/86D-17 F/G 12/4 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AFIT/GOR/ENS/86D-17

1

AD-A185 247

DTIC FILE COPY

AUTOMATING RESPONSE SURFACE METHODOLOGY FOR THE ARSENAL EXCHANGE MODEL

THESIS

Gary E. Yielding
Captain, USAF

AFIT/GOR/ENS/86D-17

Approved for public release; distribution unlimited

87 9 25 067

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY 2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GOR/ENS/86D-17	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Engineering 6b. OFFICE SYMBOL *(If applicable)* AFIT/ENS	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS *(City, State, and ZIP Code)*	7b. ADDRESS *(City, State, and ZIP Code)*
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b. OFFICE SYMBOL *(If applicable)*	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS *(City, State, and ZIP Code)*	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
AUTOMATING RESPONSE SURFACE METHODOLOGY FOR THE ARSENAL EXCHANGE MODEL (UNCLASSIFIED)

12. PERSONAL AUTHOR(S)
Gary E. Yielding, B.S., Capt, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Year, Month, Day)*	15. PAGE COUNT
MS Thesis	FROM ____ TO ____	1987 May	305

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
12	02	
14	02	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Response Surface Methodology, Arsenal Exchange Model, Sensitivity (Informational) Analysis

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Thesis Advisors: William F. Rowell, Lieutenant Colonel, USAF
Associate Professor of Operations Research

Joseph L. Faix, Lieutenant Colonel, USAF
Instructor of Operations Research

Thesis Reader: Palmer W. Smith, Lieutenant Colonel, USAF (Retired)

-- continued --

Approved for public release: IAW AFR 190-1.
LYNN E. WOLAVER 3 Sept 87
Dean for Research and Professional Development
Air Force Institute of Technology (AFC)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL William F. Rowell, Lt Col, USAF	22b. TELEPHONE *(Include Area Code)* 513-255-3362 22c. OFFICE SYMBOL AFIT/ENS

DD Form 1473, JUN 86 Previous editions are obsolete. SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

Response Surface Methodology (RSM) is automated for the Arsenal Exchange Model. The purpose is to give the using organization (AF/XOXR-MAA) the capability of rapidly answering "what-if" questions in "real time" on force structure problems. Currently, the AEM requires approximately three hours CPU time for one execution. By using RSM to build a replacement model for the AEM, a second-order polynomial equation for this thesis, the execution time is brought to within seconds. So, the RSM technique builds a replacement model for the AEM, given a specific scenario, that provides predictability within seconds. Also, using experimental designs that are othogonal and rotatable, the replacement model (second-order polynomial equation) can be used very effectively and efficiently for informational analysis (same as sensitivity analysis). Much more information can be garnered from the replacement model than the real model (the AEM). So, besides the advantage of predictability, the RSMed built replacement model provides much more informational analysis than what could have been garnered by applying sensitivity analysis to the AEM.

AFIT/GOR/ENS/86D-17

AUTOMATING RESPONSE SURFACE METHODOLOGY
FOR THE ARSENAL EXCHANGE MODEL

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Operations Research

Gary E. Yielding, B.S.
Captain, USAF

May 1987

Acknowledgments

I express my sincere thankfulness and gratitude to my wife Gail and my son Nathaniel. They hung with me and supported me through the "tough" times. They sacrificed their time and pleasure to help me often. We have grown a lot. I also sincerely thank my advisor Lt Col Rowell for his "never give up the ship" attitude. His faith in my ability to do the work kept me going many times when all I saw was gloom and despair. Also, I thank Lt Col Faix and Lt Col Smith (retired USAF) for their technical support in the RSM field. Lt Col Smith too provided help and encouragement when I needed them. Finally, I thank Professor Reynolds for his insightly comments to the mathematical portions of this thesis.

Table of Contents

page

List of Figures

Figure page

Abstract

This research is concerned with providing a means to rapidly answer "what-if" questions about force structure problems for the Air Force Wide Mission Area Analysis Division (AF/XOXR-MAA). Currently, AF/XOXR-MAA uses a deterministic computer model known as the Arsenal Exchange Model (AEM) to answer "what-if" type questions. However, three hours of wall-clock time are required for just one AEM execution.

Response Surface Methodology (RSM) techniques that include experimental designs and regression analysis are automated by Fortran-77 code. For user defined problems, this automated RSM package replaces the AEM with a second-order polynomial equation. The polynomial equation can then be used to answer "what-if" questions in "real time" through its predictability powers and/or through applied informational analysis.

An eighteen-variable (reduced to twelve variables) example is used to show how well the automated RSM package works with the AEM. The example is ran on AFIT's VAX 11/785 main-frame computer. Also, from the main-frame's software collection, the Statistical Analysis Software (SAS) package is used to verify the regression analysis portion of the automated RSM package.

AUTOMATING RESPONSE SURFACE METHODOLOGY FOR THE ARSENAL EXCHANGE MODEL

Chapter One: The Problem

Introduction

This research effort is concerned with automating Response Surface Methodology (RSM) to the Arsenal Exchange Model (AEM). The purpose is to develop a model that can substitute for the AEM. However, the alternate model will be able to more quickly produce a response while increasing the capability of informational analysis (defined in the Key Terms section of this chapter). This research endeavor is in response to a problem faced by the Air Force Wide Mission Area Analysis Division (AF/XOXR-MAA).

This chapter defines and gives some background to the problem; lists the research objectives, scope, and assumptions; and, cites the problem solving methodology. Chapter Two contains the literature review on previous applications of RSM to deterministic and stochastic models. Also, Chapter Two describes the AEM and delineates the elements of RSM. The automated methodology steps along with the programming tools that automate the methodology is outlined and explained in detail in Chapter Three. Chapter Four explains how the automated RSM works through an example AF/XOXR-MAA problem. Chapter Four presents the results of applying the methodology to the similar AF/XOXR-MAA problem, and Chapter Five closes this report with observations, conclusions, and recommendations.

Problem Background

The Air Force Wide Mission Area Analysis Division (AF/XOXR-MAA) evaluates the US strategic offensive capability by allocating strategic offensive resources ". . . to maximize strategic offensive capability in accordance with assigned mission area importance and goals" (23:2). For the evaluation, data is gathered from a deterministic model known as the Arsenal Exchange Model (AEM). The AEM allocates US strategic offensive resources according to national targeting guidance and evaluates the success of the allocations through damage expectancy calculations.

In addition to measuring the success of US resource allocations, a significant part of the evaluation is the use of informational analysis to answer "what-if" questions. Usually, informational analysis is performed, through conventional methods, by varying one AEM parameter at a time and executing the AEM for each parameter change. Therein lies the problem. According to AF/XOXR-MAA's Major Leonard, the AEM requires approximately three hours (50 minutes CPU time) to make one execution (16). Since each execution of the AEM takes approximately three hours, answering "what-if" questions is very slow. Consequently, according to Major Leonard, there is currently no way to rapidly answer predictable "what-if" questions about force structure and other parameter changes (16).

Research Problem

Automate the application of Response Surface Methodology (RSM) to the AEM to rapidly predict AEM output from approximated polynomial equations and to assist in performing informational analysis for the Air Force Wide Mission Area Analysis strategic offensive mission area.

Research Question

Can Smith's RSM variant be automated with the AEM to allow accurate predictability, give time efficient informational analysis, and produce more information from the analysis (25;26;27)?

Research Objective

Automate Smith's RSM variant with the AEM, document the automation, and then demonstrate the automation to AF/XOXR-MAA by applying the automated RSM to a strategic force structure problem.

Key Terms

Smith's RSM Variant. Smith's RSM variant is the application of RSM to deterministic models to enhance informational analysis capabilities (25;26;27). The RSM composition includes a postulated model of the true response surface (usually a low-order polynomial equation), an experimental design that explains how to gather the data necessary to estimate the postulated model, and regression analysis to perform the estimation of the postulated model's coefficients.

Arsenal Exchange Model (AEM). The AEM is a linear, goal-programming, weapon-to-target optimal allocation model (5). The AEM specifically addresses strategic force analysis and design problems (5:1-1).

Informational Analysis. Informational analysis refers to any method that generates "additional" information for the decision maker beyond that of just the solution obtained from solving the problem using baseline problem data and assumptions. Informational analysis involves examining the impact of changes to the problem composition, such as variables, coefficients, objectives, weighting factors, and/or

constraints. Applying informational analysis to a problem's composition helps a decision maker understand how sensitive a solution may be to certain problem components.

Subsidiary Objectives

Automation. Perform the necessary coding to automate the RSM variant with the AEM. "Necessary coding" involves allowing user input for problem definition, parameter coding and decoding, coded and decoded experimental design generations, AEM data collection, and regression analysis.

AEM Setup. Develop an unclassified multiple objective weapon-to-target allocation problem similar in dimension (weapons, targets, etc.) to the type used in the AF/XOXR-MAA strategic offensive mission area. This includes AF/XOXR-MAA designating the important variables (those to which informational analysis would be applied) to be used with the AEM.

Response Surface. Develop a response surface for each objective by varying several key parameters over their expected range. The number of parameters will be sufficient to make this automated RSM process useful to AF/XOXR-MAA.

Validation. Each response surface equation is an alternate model that replaces the AEM for each objective. The alternate models will estimate the AEM's responses for any given set of input. However, to insure the alternate models can be used in place of the AEM, the alternate models are validated. The response surface equations are validated by comparing their responses to responses generated by the AEM using the same random input. The alternate models can be used if the results of the comparisons are within user limits.

Evaluation. Evaluate the use of informational analysis for AF/XOXR-MAA through graphs and plots of the problem's components.

Portability. Examine the portability of this automated RSM process to AF/XOXR-MAA. To a degree, the automation objective is encompassed by portability because the type of coding and use of the coding must be compatible between AFIT and AF/XOXR-MAA computer systems.

Recommendations. Examine the possibility of further automation to include the use of other postulated models and experimental designs.

Scope/Assumptions

Problem. A problem with five objectives is evaluated. Four of the five objectives measure the results of weapon-to-target (resource) allocations through mutually exclusive target categories: leadership/communication, command, control, and intelligence; strategic nuclear; conventional; and industrial/economic. The fifth objective measures the overall resource allocation results. The measure of effectiveness for each of the five objectives is damage expectancy (DE) which is "The probability of achieving a desired level of damage, considering prelaunch survivability, weapon system reliability, weather/darkness factors, penetration probability and weapon effects" (8:11).

Problem Type. Problem composition is similar to AF/XOXR-MAA problems: resource allocation goals (damage expectancy) are setup; the allocations themselves are defined; targeting strategies and targeting priorities are employed; and the targeting constraints are listed. However, the problem will not be classified, as are the AF/XOXR-MAA problems. An unclassified problem will allow wider dissemination and application of the results of this thesis.

Factor Range. The range of each factor is fixed. If the range of a factor increases, i.e., the factor's minimum and/or maximum level changes so that the factor's new range becomes larger than its former, the entire process of generating RSM curves has to be repeated (27:62).

Extrapolation. NO EXTRAPOLATION can be performed beyond the ranges of the factor levels (27:62). The scope of the surface as described by the polynomial equation is limited by the ranges of the variables used to "fit" the polynomial equation. Therefore, it is dangerous to deduce any results outside the scope of the surface.

Automation. The automation process begins with AEM data input. The data is then organized according to the applicable experimental design. The automation continues by controlling AEM executions and collecting all AEM output. Finally, the automation is complete when a regression package uses a coded version of the experimental design with the AEM output to select the variables of the postulated model.

Assumption. Each objective's response surface can be approximated by a second-order polynomial. The second-order polynomial equation is a popular model and has previously been used with other arsenal exchange models in previous theses (i.e., Graney's, Ishihara's, and Johnson's which are discussed in Chapter Two (11;14;15)).

Abridged Methodology

A weapon-to-target allocation problem is first setup. An applicable experimental design is used to gather AEM output which is then used as input to a regression analysis technique. The output from the regression analysis is then used to construct the polynomial equations that represent the AEM. The polynomial equations are validated, and,

following a successful validation, the polynomial equations are used for AEM predictability and for informational analysis.

Conclusion

This chapter described the research problem, listed the research objectives and associated scope/assumptions, and briefly described the problem solving methodology. The next chapter reviews literature involving previous work done on the application of Smith's RSM variant to deterministic and stochastic models. Chapter Two also examines the AEM and reviews literature on the elements that constitute RSM, i.e., response surfaces, experimental design, and regression analysis.

Chapter Two: Literature Review

Introduction

First, a definition is presented on the term "Smith's RSM variant". Second, a search through earlier AFIT theses is accomplished to show how Smith's RSM variant was previously applied. Next, the AEM, the model whose output is to be "captured" through the automated RSM process, is briefly described. The components of RSM are then expounded upon so that the reader becomes familiar with the "workings" of RSM. Finally, to orient the reader, the applicability of RSM to the AEM will be shown throughout this chapter.

Smith's RSM Variant

Before describing the RSM variant, the classical use of RSM is first defined. A good definition of classical RSM is given by Douglas Montgomery:

> Response surface methodology, or RSM, is a collection of mathematical and statistical techniques useful for analyzing problems in which several independent variables influence a dependent variable or response, and the goal is to optimize this response. (20:445)

The RSM variant was first formulated and used by Smith in his dissertation (25). Later, in two separate papers (the latter an update of the first), Smith and Mellichamp explained the purpose for this RSM variant:

> . . . provides an expanded capability for conducting a more valuable analysis in a complex environment. It provides a picture of what is happening within the model being used for the study. It also provides insight into the relationship among the factors under study. "What if" analyses can be conducted economically and in real time without the necessity to obtain new computer outputs. Derived contributions of each

parameter to the value of the measure of effectiveness and ratio comparisons of the different coefficients of the model parameters provide a new measure of effectiveness for comparing the worth and capability of one system vs another within the context of the model. (26;27)

Thus, the goal of Smith's RSM variant is not to optimize a response, as is the goal of classical RSM, but rather to facilitate the use of informational analysis as applied to deterministic models.

Previous Applications

There has not been any previous work done to automate RSM with the AEM; thus, applying the RSM to the AEM through automation is the unique contribution of this thesis. Smith's RSM variant has, however, been used in previous AFIT theses. Specifically, from 1983 to 1986, nine students have applied the RSM variant in their theses. To highlight the use of Smith's RSM variant, the following paragraphs briefly explain the purpose, the deterministic model used, and how the use of Smith's RSM variant improved informational analysis for each of the former AFIT theses. The theses are recounted in chronological order.

Cooke was the first to use Smith's RSM variant (4). His objective was to develop a goal programming approach which would ". . . optimize combat power delivered to a theater during a specified time, within acceptable levels of force sustainability" (4:viii,4). Part of the goal programming approach was for Cooke to develop his own deterministic model to give ". . . a multi-objective optimization which addresses intertheater movement, intratheater movement, and deployable unit capabilities" (4:1). The other part of the goal programming approach was the use of the RSM variant to enable full parametric informational analysis (4:viii). Cooke praised the RSM variant in his conclusions:

> . . . it has been demonstrated that a flexible response surface methodology can result in a reduction of a fully computerized and intricate large scale model to an equation which can be programmed on a hand-held calculator, and with which major force design concepts may be rapidly searched, with minimal probable error. (4:130)

Next, Graney applied Smith's RSM variant to an aggregated strategic nuclear exchange model with a non-commensurate, two-objective problem (11). His problem was to show the optimal trade-offs between the responses to the two objectives over their entire operating domain (11:vii). The objectives were to maximize the damage expectancy for both counter-force and counter-value targets. Graney also reported the advantages of increased informational analysis from using Smith's RSM variant:

> The methodology's advantage over other multicriteria decision theories is its ability to consider large numbers of alternatives, show the MOE's comparisons throughout the range of alternatives, and use non-commensurate measures of effectiveness. (11:79)

Manacapilli used Smith's RSM variant for the purpose of ". . . the identification of cost effective strategic force mixes" (18:vii). For his work, Manacapilli used an arsenal exchange model as his deterministic model. Manacapilli postulated that an economic production function could capture the output of the deterministic model and, coupled with Smith's RSM variant, increase informational analysis:

> . . . economic production functions can be easily fitted to the response surface of a deterministic model and that the information gained from the fitting of these economic production functions can provide insight into the production process that will be useful to a decision maker. (18:6.4)

Meitzler applied Smith's RSM variant to AFALC's Budget/Readiness Analysis Technique (BRAT) model (19). BRAT is a deterministic simulation model that converts levels of logistic resources into a sortie

rate (19:vii). Even though BRAT is relatively quick to operate, no informational analysis had been applied to it, so Meitzler's problem was to extract informational analysis from the BRAT (19:22). Therefore, the purpose of Meitzler's thesis was to improve BRAT's ability for ". . . efficient multi-dimensional sensitivity [informational] analysis" (19:vii). Meitzler concluded that the fitted response equation ". . . provides an approximate answer to a 'what-if' question without using the time and computer resources necessary to rerun BRAT" (19:103).

Sparrow applied RSM in a more classical approach to explicitly express the relationships between the input and output factors of a simulation program (28). The purpose of her thesis was to facilitate the use of sensitivity (informational) analysis between the input and output factors of a simulation model (There has been some work in applying RSM to stochastic models, such as simulation models, for several years but specifically to optimize simulation output rather than to fully describe the surface). Sparrow's method was to automate RSM as a user-friendly computer package that could be attached to any Fortran based simulation model (28:4). The usefulness of Sparrow's application was that it revealed the relative importance of the simulation model's input factors to its output and illuminated the interdependence between pertinent factors within the simulation model (28:3).

Libby's thesis dealt with determining ". . . the cost of withholding ICBMs from launch until after detonation of Soviet missiles, even if there is warning of an attack" (17:1-13). Libby's analysis was threefold:

> First, a relative measure of military effectiveness and cost for the current US ICBM force will be determined as a baseline case. Force modernization options are then investigated

> for their contributions to effectiveness and costs. Second, the trade-offs between modernizing the force structure with new weapon systems and a launch before impact policy are studied. At this point, a method for computing the cost of withholding a missile that is under attack is presented and sample calculations are made. Third, a discussion of the uncertainties and risks of a launch before impact policy, in light of the cost savings incurred, conclude the analysis. (17:1-15)

Just like Graney and Manacapilli, Libby also used an arsenal exchange model as his deterministic model. Also, as Graney and Manacapilli ascertained, Libby discovered how useful Smith's RSM variant is for informational analysis:

> The use of response surface methodology in conjunction with deterministic models [Smith's RSM variant] is the key to analysis of a launch before impact. Without it, the conventional approach of making a limited number of runs on an arsenal exchange model would not provide such a detailed description of US ICBM effectiveness. Indeed, the methodology condenses the results of specifically defined experiments into one equation which can be manipulated as desired. (17:7-1)

Johnson also applied Smith's RSM variant to an arsenal exchange model (15). Johnson used a deterministic model called the Allot computer program: a linear programming based arsenal allocation model used by SAC analysts (15:vi,72). The purpose for Johnson's thesis was to develop a substitute model, in equation form, for the arsenal exchange model that insured accurate predictability and increased informational analysis:

> . . . to capture the output of this nuclear exchange model [ALLOT] into a single equation which provides the relative contributions to the overall method of effectiveness of any of the factors chosen and permits efficient multi-dimensional impact analysis of the selected model parameters. (15:vi)

Johnson's results, as favorable as previous results on the use of Smith's RSM variant, promised increased awareness of force structure and employment concerns:

> This capability [Smith's applied RSM variant] will allow analysts to investigate the nature of this model and explore the response surfaces generated in force structure analyses. The model [with the RSM improvement] should provide a new dimension of insight into force structure and force employment issues and allow analysts to answer questions concerning the relative importance as well as the interaction of the parameters in the problems that are being modeled. (15:73)

Donovan applied Smith's RSM variant to a macroeconometric deterministic model (9). The purpose for Donovan's thesis was ". . . to apply response surface methodology [Smith's RSM variant] to a macroeconometric model to facilitate better analysis with the model" (9:viii). From the response surfaces that represent the output to the macroeconometric model, Donovan derived multipliers ". . . to characterize the static and dynamic properties of econometric models and to conduct policy simulations" (9:88). As with previous theses, Donovan also had success with Smith's RSM variant:

> The advantage of response surface derived multipliers over multipliers derived by model linearization is that the model does not have to be linear or near linear for multipliers to be valid over a wide range of variable values. In addition, significant interactions and higher order effects can be identified. The advantage of response surface multipliers over control-disturbed multipliers is that they more completely characterize the relationships in the model and reduce the number of runs required to estimate multipliers. (9:88)

Finally, Ishihara investigated the possibility of improving Smith's RSM variant (14). Ishihara's problem was to compare experimental designs for efficiency (14:1-8). The experimental designs were of two types: bias minimizing and variance minimizing designs. Bias minimizing designs try to minimize the error between the postulated surface and the actual surface over a region while the variance minimizing designs try to minimize error due to sampling (14:1-2). Because deterministic models have no sampling error, only bias minimizing designs,

not variance minimizing designs, would be used with Smith's RSM variant. However, according to Ishihara, ". . . Manacapilli [18] showed that a bias minimizing design required more design points than a variance minimizing design while gaining only a small improvement in the fit of the postulated surface" (14:1-8). Ishihara used a deterministic nuclear exchange linear programming model to show that ". . . there were no significant differences between the two types of experimental designs, provided that the designs maintained certain characteristics, such as, rotatability and orthogonality" (14:4-4). (The terms "rotatability" and "orthogonality" are explained later in this chapter.) However, Ishihara also reported that:

> . . . more design points improves accuracy, provided the points are spread throughout the entire surface. Similarly, if the region of interest is spherical, a rotatable design improves the accuracy of the response equation. (14:4-2)

Therefore, even though no significant differences were found between bias and variance minimizing designs, the effectiveness of a response equation may depend on which experimental design requires more design points (14:4-2).

Previous Work Summary. The preceding account of theses that have applied Smith's RSM variant have all reported the usefulness of the RSM variant to informational analysis. First, time, an important factor for any analyst when performing informational analysis, was decreased because extra runs of the deterministic/stochastic model, due to conventional methods of informational analysis (vary one parameter at a time syndrome), were not necessary. Second, by using the RSM variant as opposed to conventional informational analysis methods, much more information was generated: the relative worth of each factor, the contri-

bution to the response per a unit of each factor, etc. (27:63). (These two advantages are demonstrated in Chapters Four and Five.)

The previous theses also proved that arsenal exchange models could be used as the deterministic model in the RSM variant. Of the nine theses, five of them used an arsenal exchange model as their deterministic model. And, of those five, Johnson's work is most closely related to the effort put forth in this thesis. But, it is the combined success of these prior theses with arsenal exchange models that provides a basis for applying the RSM variant to the AEM.

The AEM

As defined in Chapter One, the AEM is a linear, goal-programming, weapon-to-target optimal allocation model (5). Furthermore, "The AEM is an aggregated, two-sided strategic exchange model with a diverse set of scenarios and analyst controls" (6:2).

The Martin Marietta Corporation originally developed the AEM in 1965 to help understand the TITAN weapon system (6:2). However, since its conception, the AEM ". . . has developed into a general purpose force analysis model" (6:2). Today, the AEM ". . . has become one of the most widely used strategic force analysis models in the defense community" (6:2).

The AEM is used for many diverse types of strategic force analysis problems: strategic weapon system analysis, strategic nuclear policy support, arms control, force management, intelligence support, and/or general strategic calculations (6:2). For any type of problem, "The AEM performs optimal allocations for either side in a scenario and evaluates damage attained by the allocations" (6:2).

The AEM uses the dual variables that are always associated with linear programming to solve problems (6:22). The dual variables, or mathematically, the lambdas, allow the AEM to decompose its problems into a sequence of smaller, more manageable linear programs (6:22). This decomposition decreases the solution time and produces the same results as would have been produced if the problem had been solved as a single, gigantic linear programming problem (6:22). (Note: It is beyond the scope of this thesis to explain duality theory. The dual variables are only mentioned to inform the reader about the process that AEM uses to solve its problems. More indepth information on duality theory can be obtained in Chapter Three of Hillier and Lieberman, Chapter Eight of Ignizio, and/or Chapter Four of Wu and Coppins (12;13;30).)

Input to the AEM is very flexible. The AEM allows for multiple-objective type problems to be inserted as goals. The goals are solved preferentially according to user-defined priorities. Scenario-type input is mainly in the form of weapons and targets. The AEM receives weapon and target inputs ranging from the number of weapons and targets available to weapon and target system characteristics (i.e., weapon reliability and target hardness). Constraints are received in the form of weapon-to-target allocation rules and strategies and information uncertainties. AEM's characteristic of flexible input is a prominent reason why the AEM is a widely used strategic force analysis tool.

The AEM has the capability of producing vast output. But, the amount of output generally depends on what type of problem the AEM is solving. Predominately, though, the optimal weapon-to-target allocations and the AEM response (damage expectancy) to any or all objectives are almost always desired.

One of the quickest ways to learn the ABM is to apply its user's manual to specific problems. The ABM's users' manual has been updated to reflect the 1985 revision to the ABM (6). To help understand the ABM algorithm, texts on linear/goal programming by Hillier and Lieberman, Ignizio, and Wu and Coppins provide excellent starts (12;13;30).

Response Surface Methodology

The methodology of Smith's RSM variant is to fit a response surface to the output of a deterministic model. Intrinsic to Smith's RSM variant and classical RSM are experimental design techniques and regression analysis. Experimental design techniques explain how to collect data from a model, deterministic or stochastic, while regression analysis uses the collected data to fit and/or allow inferences to be made about the postulated model. More on RSM can be found in Box and Draper, Montgomery, or Myers (3;20;21): Box and Draper have recently combined their efforts to produce a book with some excellent chapters on the components of RSM; Montgomery devotes one chapter to the purpose and use of RSM; and Myer's text is the only text devoted entirely to RSM.

Assumptions. There are a few assumptions upon which RSM is based. Myer's text has concisely listed these assumptions:

> 1. A structure $\eta=f(X_1,X_2,\ldots,X_k)$ exists and is either very complicated or unknown. The variables involved are quantitative and continuous.
> 2. The function f can be approximated in the region of interest by a low-order polynomial . . . .
> 3. The independent variables $X_1,X_2,\ldots,X_k$ are controlled in the observational process and measured with negligible error. (21:62)

Response Surface. Probably the crux of RSM is the assumption that the structure of the output for the deterministic or stochastic model can be postulated by a low-order polynomial equation. The output of any

model can be viewed as a k+1 dimensional surface if all responses to increment variations in the input variables were gathered and plotted in k+1 space (k is the number of input variables). Polynomial equations can usually be thought of and sometimes pictured as a surface. In three dimensions, especially if enough of the output is collected and plotted, a surface (response surface) would be visible and this surface would be mathematically explained by the polynomial equation.

The most common type of low-order polynomial equations are those of either first or second order. These two types of equations can be found in almost any statistical reference book:

$$Y_{lm} = \beta_0 + \sum_{i=1}^{k} \beta_i X_i + \sum_{\substack{i\,j \\ i<j}}^{k\,k}\sum \beta_{ij} X_i X_j + \varepsilon_{lm} \qquad (2.1)$$

$$Y_{lm} = \beta_0 + \sum_{i=1}^{k} \beta_i X_i + \sum_{\substack{i\,j \\ i<j}}^{k\,k}\sum \beta_{ij} X_i X_j + \sum_{i=1}^{k} \beta_{ii} X_i^2 + \varepsilon_{lm} \qquad (2.2)$$

where

k represents the number of independent input variables.
l = 1, 2, 3, . . . , N where N represents the total number of observations (responses from the model).
m = 1, 2, 3, . . . , n where n represents the total number of replications at an observation.

Equation 2.1 is a first-order polynomial equation while equation 2.2 is a second-order one. For both equations, the response term is the variable located on the left side. The first term on the right side of both equations is the intercept. To the right of both intercepts are the linear terms. They are shown grouped together by a summation symbol. Linear terms represent the main effects of the equation. Following both sets of linear terms is the summation of cross terms representing the cross effects of the equation. (The first-order

equation (2.1) need not contain the cross terms if the user doesn't believe interaction exists.) For equation 2.2 only, the last set of terms is the summation of second-order terms, each representing a pure quadratic effect for the second-order polynomial equation. The last term for each equation is the error term associated with each response. The error term is a very influential part of the equation for it can control the experimental design selection, and/or it can be used to judge the fit of the polynomial equation.

Invariably, when choosing an alternate model to postulate the real model, of high importance is the need to obtain as much information as possible from that postulated model. Not only should the postulated model accurately predict the real model, but it should also contribute additional information through informational analysis. Therefore, the choice for the postulated model is important because not all postulated models are amenable to informational analysis. The polynomial equations of 2.1 and 2.2, though, are excellent choices for postulated models because they are very conducive to informational analysis provided they are constructed with experimental designs that insure rotatability and orthogonality between the model's terms (refer to the Experimental Designs subsection of this chapter).

Equation 2.1 or 2.2 is called the "postulated" model when either is used to represent the output of some "real" model. The choice of which equation to use is dependent upon the user's understanding of the real model and/or the eventual use of the postulated model. But, once an equation is picked as the postulated model, the coefficients of that model will need to be estimated. The first step towards estimating the coefficients is the selection of an experimental design.

Experimental Designs. The estimation of the postulated model's coefficients is actually done through multiple regression, which is explained later in this chapter. However, multiple regression requires data, and this data has to be obtained from the real model. But, how much data is enough; what should the values of the model inputs be; and how many replications, if any, should there be?

Experimental designs were developed to answer those questions and to fulfill other purposes. Experimental designs explain how much and from which model variables data should be collected. In doing so, though, experimental designs also fulfill another function: results and conclusions drawn from informational analysis are correct and valid (20:2). So, experimental designs not only explain how much data to collect, but they also insure the collection is correctly accomplished so that any analysis of the fitted model will give valid and objective results (20:1-4).

By explaining "how" the data should be collected, the experimental design is actually minimizing the expected mean square error averaged over the response surface (2:622). Error naturally occurs as a result of using the postulated model, but, error can also occur from the postulated model's formulation (2:622). The first type of error, called bias error, occurs when a postulated model is used in place of the actual model. Variance error, the second type of error, can occur from the data that is collected to estimate the coefficients of the postulated model. An experimental design attempts to collect data from the real model so as to minimize the magnitude of the error terms.

For equations 2.1 and 2.2, error is accounted for in the error terms associated with each response, i.e., ε_{im}. For any input, this

error term is the difference between the real response (η) and the postulated response (Y). And, it is this error term that contains the two components of error: variance and bias. The variance error occurs from sampling the same data point, i.e., the different responses that occur from the same set of model input. The bias error appears when the postulated model's surface is not an exact copy of the real model's surface, i.e., the error between the postulated and real response surface. So, an experimental design's objective, when explaining how to collect data from the actual model for the purpose of estimating the postulated model (equation 2.1 or 2.2), is the minimization of the expected mean square of the error term associated with the postulated model, i.e., the ε_{im}.

There are many types of experimental designs. They are labelled by their purpose, use, and characteristics. Since error is composed of bias and/or variance, experimental designs are classified according to the type of error they minimize: bias-minimizing designs, variance-minimizing designs, and designs that try to minimize both types of error for any type of model (29). Experimental designs are also grouped by their use, i.e., the type of experiment or situation they are used for. Last, experimental designs are also classified with respect to their characteristics such as efficiency, orthogonality, and rotatability.

Variance-minimizing designs are usually appropriate when the user believes that there is more variance error than bias error. Conversely, bias-minimizing designs would probably be appropriate if the user believes that there is more bias than variance error or if the model is deterministic (no variance error). However, Ishihara points out that these assumptions alone might not suffice for design selection (14).

Experimental designs have very wide and diverse usage. They are employed with agriculture, chemistry, advertising and sales, social statistics, and in many other areas. The model can be stochastic or deterministic and the data can be empirical or computer generated. The use for experimental designs are limitless.

The characteristics of a design can be crucial to the design selection. Of the characteristics noted above (efficiency, orthogonality, rotatability), design efficiency might be the most decisive characteristic because resource availability constricts the entire experiment. For instance, if computer time is very expensive or almost nonexistent, the design efficiency for this example (required number of computer runs) would probably be the determining factor for design selection. Another resource availability example might be the abundance of proper weather and/or land for an agricultural experiment. Likewise, for a chemical experiment, the amount of chemicals and their costs when required for a chemical experiment would be important in selecting a design. Resources become scarce, either naturally or otherwise, therefore the ability of a design to make the most of what is available (efficiency) can be the most important criteria for design selection.

Besides efficiency, there are two other important design characteristics: orthogonality and rotatability. These characteristics pertain to how easy information is obtained, through informational analysis, from the postulated model, and to how reliable the information is when extracted anywhere over the range of the problem.

Orthogonality is the property that ensures the data collected by the experimental design will lead to independent factors for the response equation (7:587). The independence of the factors will allow

information to be obtained more easily through informational analysis. Chapter Five shows some of the informational analysis results taken directly from the response equations containing independent terms.

Orthogonality can be accomplished by "coding" the real model's input. The actual input is scaled to numbers that satisfy the mathematical requirement for orthogonality (independency between all columns of the design matrix) (3:20,21;21:43,166-174):

$$X_i' = (X_i - A_i) / B_i \tag{2.3}$$

where

X_i' is the coded variable.
X_i is the actual variable.
A is the midpoint between the highest and lowest value for the actual variable.
B is half the difference between the highest and lowest value of the actual variable.
i = 1, 2, 3, . . . , k where k represents the number of independent input variables.

The values of the coded variable depend on the postulated model. For example, for equation 2.1, the coded values will be +1 and -1. And, for equation 2.2, the coded values will be +1, 0, and -1.

The rotatability feature occurs when ". . . the variance of the predicted response (Y) at some point (X) is a function only of the distance of the point from the design center, and not a function of direction" (20:355). Rotatability allows the user to be justly confident with any information drawn from the postulated model because ". . . points in the factor space which are the same distance from the origin are treated as being equally important" (21:165). Complete confidence in the postulated model, along with the ability to easily obtain information from the postulated model, makes rotatability and orthogonality desirable design characteristics.

Many designs are available for many uses. The type of experiment, the resources available, and the user's requirements dictate what design is chosen. Additional information on experimental designs can be found in texts by Box and Draper and by Davies (2;7). A good summary is afforded by Steinberg's article which reviews the principles and methods of experimental designs (29). However, Montgomery is highly recommended because he expertly shows the binding relationship between experimental designs and statistical analysis, specifically, multiple regression (20).

Multiple Regression. Multiple regression refers to regression analysis applied to a linear equation with two or more independent variables (22:226-263). Regression analysis uses the method of least squares ". . . to examine data and to draw meaningful conclusions about dependency relationships that may exist" (10:4). By using the data collected from the AEM as prescribed by the experimental design, the method of least squares will try to prove that the relationship as shown by equation 2.2 exists between the AEM weapons and the AEM response of damage expectancy.

Multiple regression is a valuable statistical tool for making inferences about the postulated model (equation 2.2), for obtaining estimations to the model's coefficients, and for aiding in model construction and validation. However, since a deterministic model contains no variance error, many multiple regression statistics cannot be used. Therefore, multiple regression is not reviewed in detail; only the method of least squares and the regression statistics used in automating RSM with the AEM is reported.

The main use for multiple regression, one most relied upon by Smith's RSM variant, is the parameter coefficient estimations of the postulated response surface equation (the βs in equations 2.1 and 2.2). This is done by the least squares method (22:238-239). The least squares method minimizes the sum of the squares of the deviations (statistically called residuals) between the postulated model's output and the true model's output (21:23). The sum of the squares of the deviations is called the error sum-of-squares (SSE). In the process of computing the SSE, the postulated model's coefficients are estimated.

The SSE is called a statistic and can be used, along with other statistics, to make inferences about the estimated postulated model. Again, the SSE is the total squared difference between the actual model (η) and the postulated model (Y) with respect to the same data design points. The error term in equations 2.1 and 2.2 (ε_{im}) reflects the difference between the postulated model and the real model for any design point. In computing the SSE, the error associated with each design point is squared and summed with all other errors from all other design points (N) replicated n times:

$$SSE = \sum_{i=1}^{N} \sum_{m=1}^{n} (\eta_{im} - Y_{im})^2 = \sum_{i=1}^{N} \sum_{m=1}^{n} (\varepsilon_{im})^2 \qquad (2.4)$$

The lower the SSE value, the better the postulated model fits the actual model at the design points. Usually, though, the value of the SSE statistic decreases as the number of terms in the postulated model increases. Therefore, it is advisable to evaluate the SSE statistic with regards to other model statistics like the R^2.

The R^2 statistic allows inferences to be made about the postulated model's explainability of the real surface. This is done in percentile

values, and the higher the value, with 1.00 representing a perfect explainability, the better the postulated model. If an R^2 statistic has a high value, the variables of the postulated model explain a large proportion of the total variation about the mean response (10:33). If a postulated model, whose coefficients have been estimated, has a high R^2 value, much confidence can be placed in that postulated model's ability to replace the actual (real) model. If the R^2 value for an estimated postulated model is low, the postulated model should not be used in lieu of the true model.

Like the SSE, the R^2 statistic is fallible. The value of the R^2 statistic slightly increases with each addition of a term to the postulated model. So, a postulated model, that accurately represents the real model, with many terms will probably have a high R^2 value. Subsequently, conclusions ascertained on model adequacy and/or model explainability should so be done by evaluating the R^2 and SSE statistics together, with other statistics, such as residuals, and with personal experience of the problem.

The residual is the statistical name given to the error term of the postulated model ($\hat{\epsilon}$). A residual is the deviation between the predicted (postulated) model's output and the real model's output for the same single data point. The fitting of the predicted model is done with a finite set of data design points. Each design point will have a residual.

The SSE and R^2 statistics are good indicators for overall model adequacy and explainability. But, a model may be better at certain points and worse at others. The residuals help show where the postulated model better predicts the real model and where it doesn't.

Residuals are also helpful for validating the postulated model. Once a postulated model is estimated, residuals can be computed from a set of randomly chosen data points. The residual's mean and standard deviation from the randomly chosen data points will help in assuring the predictability accuracy of the estimated model.

One other regression statistic is used for model construction: the sum-of-squares for each model parameter. The parameter sum-of-squares show the individual strength of each model parameter. The higher the sum-of-squares for a particular parameter, the likelihood that parameter belongs in the postulated model. Likewise, the lower the sum-of-squares for a particular parameter, the likelihood that parameter doesn't belong with the postulated model. A model's parameter sum-of-squares allows the user the ability to delete non-contributing parameters, and, thus, reduce a full estimated model to a more manageable reduced form.

Multiple regression is a valuable and necessary element of the RSM variant. Multiple regression completes the fitting of, allows inferences to be made on, and helps validate the postulated model. All of this is done through the least squares method. For more information on the least squares method along with explanations on the model statistics, refer to texts by Draper and Smith or Neter, Wasserman, and Kutner (10;22).

<u>RSM Summary</u>. In summary, the real response (η) to some experiment or from the output of some model, etc., depends on the input values of certain independent variables (X_1, X_2, X_3, etc.). By applying RSM to the experiment or model, the user is trying to capture the response in a surface through an approximating equation, usually a low-order polynomial (21:61-66). For this thesis, the response to be modelled is the

output from the AEM, and the purpose is AEM predictability and increased informational analysis. The latter purpose is due to the use of deterministic models and certain experimental designs.

In practice, a response surface is first postulated, and then multiple regression is used to develop a regression equation that "captures" the postulated response surface. To develop input for estimating regression coefficients, different sets of AEM weapon-to-target allocations are set up, and the data generated by these allocations (optimal expected target damage) are collected. An experimental design is used to select the various combinations of weapons to be allocated in order to build the response surface. So, the experimental design determines the data points to be collected, AEM executions produce the data, and multiple regression uses the data to fit the postulated response surface equation.

Conclusion

A definition was developed on the term "Smith's RSM variant". Smith's RSM variant is RSM applied to a deterministic model, mainly for the purpose of increased informational analysis. Second, former thesis applications of Smith's RSM variant were concisely listed. The AEM was next described through its uses and characteristics. Fourth, the components or RSM were explained to give the reader a working knowledge of RSM. Finally, to orient the reader, the applicability of RSM to the AEM was shown throughout the chapter. The next chapter shows how the RSM variant is incorporated in the methodology used to solve the research problem defined in Chapter One.

Chapter Three: Methodology

Introduction

This chapter describes the methodology and then the computer software tools that automate Smith's RSM variant with the AEM. The methodology is described in the step-by-step sequence a user should use when applying the automated Smith's RSM variant to an AEM problem.

Methodology Steps

AEM Setup. The first step of the methodology is for a user to understand what is to be modelled: what AEM inputs are of interest, and over what range are they of interest? (Remember that AEM inputs include weapons and/or their characteristics, such as probability of arrival, circular error probable, etc.) The answer to those two questions are vital since the entire methodology is an outgrowth of fitting the postulated equation around the AEM inputs. Therefore, a user should first pick the AEM inputs (the variables of interest), including their range of values, and, then, set up the AEM weapon-to-target allocation rules and constraints (targeting strategies and priorities). The AEM inputs become the variables in the postulated model.

Model Postulation. The second step is to postulate a model that will "capture" the response of the AEM. Restating the RSM assumptions (listed in Chapter Two) with the AEM, the output from the AEM (damage expectancy) has some "true" structure (η) that can be "approximated" by a low-order polynomial (Y) whose independent variables (X_1, X_2, . . . , X_k) are the AEM inputs (number of weapons, probability numbers, weapon accuracy numbers).

For the AEM problem, it is assumed that each of the problem's five objectives (listed in Chapter One) can be approximated by a second-order polynomial equation (equation 2.2). The assumption is based upon previous successful applications of Smith's RSM variant to other arsenal exchange models (11;14;15;17;18). Thus, equation 2.2 becomes the postulated model for each objective of the AEM problem. The response (measure of effectiveness) for each of the five objectives is damage expectancy, and each objective's goal is to maximize the response.

The Experimental Design. After postulating a model to represent the AEM response surface, the next step is to pick an experimental design. To recapitulate, an experimental design explains how to collect data from the AEM for the purpose of "fitting" the response equation. Since the AEM is a deterministic model, there is no variance error due to sampling. Therefore, the only error component that needs to be minimized is bias. A bias-minimizing design would probably be chosen, but because of Ishihara's work (highlighted in Chapter Two), variance-minimizing designs will be used instead (14).

Many variance-minimizing designs are available for second-order equations but, for this thesis, only one type is used with the AEM: Box-Behnken designs (Appendix A). Box-Behnken designs are picked over other designs, such as Central Composite or factorial designs, because they are much more efficient (less AEM executions needed), especially when the number of design variables become large. Another major determining factor for choosing Box-Behnken designs is that they are proven. As described in Chapter Two, Box-Behnken designs have successfully been used with other arsenal exchange models. Thus, efficiency and their proven nature are the chief reasons for selecting Box-Behnken designs.

The Box-Behnken designs also have the desirable properties of rotatability and orthogonality, at least they are nearly orthogonal. Because the intercept term is correlated with the pure quadratic terms which are correlated with each other, the Box-Behnken designs are not completely orthogonal. However, orthogonality, and, thus, term independency, is preserved within the main effects, cross effects, and between the two. Therefore, for informational analysis, Box-Behnken designs are very productive. The Box-Behnken designs are shown in Appendix A, and an article by Box and Behnken explains their construction (1).

As Appendix A explains, Box-Behnken designs are available for three to sixteen variables inclusive, but are not complete: there are no Box-Behnken designs for eight, thirteen, fourteen, or fifteen variables. To overcome this problem, three solutions exist: combine similar variables, use dummy variables, or apply AEM variable controls. The first solution combines variables of similar characteristics. For example, two ICBMs are very similar in every feature except name. If the AEM problem contains thirteen variables of interest for which no Box-Behnken design exists, those two ICBMs can be combined to bring the variables down to twelve. A Box-Behnken design exists for twelve variables. This solution is recommended if computer time is not abundant, or if "time to complete the project" is short. However, the combined variables solution is not recommended if informational analysis is highly desirable for each of the combined terms separately, because any information gathered from informational analysis applies to both variables equally since they are combined.

The second solution involves inserting dummy variable(s) in the AEM setup and, subsequently, in the postulated model. A larger design is

required, though, and, thus, more AEM executions on the computer. But, if informational analysis is desirable for all variables, the second solution is recommended.

The two previously discussed variable-reduction techniques are mutually exclusive. However, the third variable-reduction technique, AEM control variables, can be used with either. The AEM control variables technique uses a postulated variable (design variable) to control more than one AEM variable of interest. For example, a bomber can carry two types of bombs. Instead of using two design variables, one for each bomb, use one design variable to represent the aircraft that carries both types of bombs. This technique along with the combine similar variables technique are used to help solve the AEM problem for this thesis. Their application to the AEM problem is reported in Chapter Four.

Data Collection. After the experimental design is chosen, data has to be collected from the AEM for the purpose of estimating the postulated model's variable coefficients. This step is the application of the experimental design to the AEM setup, that is, executing the AEM the required number of times with the AEM input modified by the experimental design for each execution. The data collected from the AEM are the responses to each of the five objectives for the AEM problem.

Regression Analysis Application. This step applies the collected data from the last step to a statistical regression package. The regression package estimates the postulated model coefficients. Once the variable coefficients are ciphered, the postulated model becomes the estimated model, specifically the full estimated model since the estimated model will contain all the estimated variable coefficients.

The full estimated model is used to predict AEM responses. The AEM input values are applied to the full estimated equation instead of the AEM. The result is an estimated (predicted) response to what the AEM would have given. The full estimated equation is the appropriate model to use for AEM predictability, but, should probably not be used for informational analysis, because the full estimated model contains all the variable terms, especially the non-orthogonal ones, that would make informational analysis cumbersome.

For informational analysis, the full estimated model should be reduced, if possible. To help reduce the full estimated model, the variables' sums-of-squares are utilized. The variable's sum-of-squares help determine which variables affect the response the most: how well the response is explained when the variable is varied. Those variables that do not or only slightly affect the response when compared to other variables are deleted from the full estimated model.

Those variables that contribute nothing to the response can be immediately eliminated. However, a cutoff has to be determined for excluding all other variables that only slightly affect the response. The cutoff is subjective. For this thesis, the cutoff is arbitrarily set at five percent of the total sum-of-squares for regression (SSR). Therefore, all those variables that make up less than five percent of the SSR are omitted. This, of course, reduces the estimated model to only those variables that affect the response the most.

Before the full, and, thereby, the reduced, estimated models are used in lieu of the AEM, they should be shown to adequately represent and accurately explain the AEM. Also, the estimated models should be validated to assure the user the estimated models represent the entire

AEM response surface. For these reasons, the regression package also computes regression statistics other than the parameters' sum-of-squares. In addition, the regression package computes the SSE, R^2, and residual statistics. As more fully explained in Chapter Two, the additional statistics help the user evaluate the estimated model's adequacy, explainability, and validity.

Decode Coefficients. The postulated model's variable coefficients are estimated using coded values for the AEM input (equation 2.3). Therefore, if the estimated models, full and reduced, are used with the coded coefficient estimates, predictability and informational analysis are hampered, because, to use the estimated models, all AEM input would have to be coded. So, to enhance predictability and informational analysis, the estimated variable coefficients are decoded.

Postulated Model Completion. After the variable coefficients are decoded, the construction of the postulated model is completed by inserting the decoded variable coefficients into their respective places within the postulated model. Once the postulated model is completely built, it is referred to as the estimated model or estimated equation.

Estimated Model Validation. The estimated model is checked for validity by comparing its output to AEM output using twenty randomly chosen, previously unused data points. The residual statistic from multiple regression is used to aid in the validation process (discussed in Chapter Two). The residuals are summed, and then their mean and standard deviation are computed. Evaluation of the results is at the discretion of the user. The user should know what limits are acceptable for the residuals' mean and standard deviation.

Methodology Tools

Two computer tools are developed to implement the methodology. The first, Tool One, is written in Fortran 77. Tool One has two functions. First, Tool One collects, organizes, and develops all the necessary input for AEM executions. Second, Tool One aids the user in constructing and validating the postulated/estimated model.

The second tool, Tool Two, is written with digital commands from the Digital Command Language (DCL) of the VAX/VMS 11/785 computer. Tool Two's purpose is to execute the AEM the required number of times and collect the AEM output in a specified format for input to a regression package.

Tool One. Fortran 77 was chosen as the computer language for this programming tool because Fortran is a widely used language in the Air Force. Appendix B shows the Fortran code and Appendix C contains the User's Guide for Tool One. By coding Tool One in Fortran 77, the accomplishment of the portability subsidiary objective (from Chapter One) is attempted. Tool One consists of five major parts: data input, coded design, uncoded design, regression application, and coefficient decoding.

Data Input. The user is queried via the computer terminal for all data input that relate to the AEM variables of interest. The user is asked to input interactively the number of design variables, their names (MMII, B-52, Trident C4, etc.), their ranges, and the number of AEM input variables each postulated variable controls. (i.e., a B-52 postulated variable can control two AEM input variables because a B-52 can carry both gravity bombs and SRAMs at the same time.) The data is checked internally for program limitations and "common sense" mistakes

as the user inputs it. If a mistake occurs, the user is asked to reenter the last input.

Uncoded Design. This section of the code takes the user input and applies the applicable Box-Behnken design to produce a data matrix containing AEM inputs. Each row of the data matrix represents one execution of the AEM. The AEM input data in the data matrix is uncoded, and, thus, represent values for all the AEM variables of interest, only adjusted per row by the experimental design. Once the uncoded design is built, Tool Two applies it to the AEM for data collection.

Coded Design. After Tool Two is finished with AEM execution and data collection, Tool One is re-executed. The coded design section of Tool One is used to produce the same data matrix as in the previous section, but in coded values, i.e., 1, 0, and +1 (refer to equation 2.3). The coded data matrix and the AEM output collected by Tool Two are used as inputs to the regression package.

Regression Application. The coded data matrix (experimental design) and the collected AEM data (from the experimental design) are used as inputs to a regression subroutine, within Tool One, for the purpose of estimating the postulated model's coefficients. Also, the sum-of-squares for each estimated model variable is computed to help the user reduce the full estimated model, if so desired. Furthermore, the SSE, R^2, and residual statistics are calculated to help the user check the estimated model for accuracy, explainability, and validity. The regression subroutine is included in Tool One to help make the RSMed-AEM methodology more portable and flexible. If a more advanced, dedicated regression package(s) is/are acquired (e.g., SAS, BMDP, or SPSS), Tool One's regression subroutine need not be used.

__Decode__. The Decode section is used after the regression package has estimated the postulated model's variable coefficients. The input to the Decode section are the beta coefficient estimates from the regression package. The output from Decode is the estimated decoded coefficients for the postulated model's equation.

__Tool Two__. The DCL commands that make up Tool Two are part of the VAX/VMS command language that allows the user to converse with the 11/785 (24:2-22). Consequently, Tool Two may not be as portable as Tool One since the DCL commands are peculiar to the VAX/VMS system. The DCL commands that make up Tool Two's program are shown in Appendix D.

The purpose for Tool Two is to regulate the executions of the AEM. The output from the Uncoded Design section of Tool One is the input to Tool Two. The output from Tool Two, the collected AEM responses, is used as input to the Coded Design and Regression sections of Tool One.

Conclusion

The automated RSMed-AEM problem-solving methodology involves developing an AEM scenario, postulating an alternate model, choosing an experimental design, collecting AEM data, applying regression analysis for coefficient estimation and model examination, decoding the estimated model's coefficients, completing the estimated model's construction, and validating the estimated model.

Most of the time-consuming steps have been incorporated into two computer programs. The first computer program (Tool One) collects the data characteristics for the AEM variables and then supplies the second computer program (Tool Two) with a coded data matrix (experimental design). Tool Two uses the coded data matrix to control AEM execution

and data collection. Next, Tool One takes the AEM data collected by Tool Two and uses it with a coded version of the AEM data matrix as input to a regression subroutine within Tool One. Tool One uses the output from the regression package (the variable coefficient estimates) as input to the Decode section of Tool One. The output from the Decode section (the decoded variable coefficient estimates) is then inserted into the postulated model equations creating the estimated model. Finally, the estimated model is checked for validity by comparing its output to corresponding AEM output.

The next chapter further explains the methodology steps through an application. Detailed use of the programming tools are highlighted so the user can become easily familiar with their use.

Chapter Four: An Application

Introduction

This chapter reports an application of the RSMed-AEM methodology outlined in Chapter Three to an AEM problem/scenario. The objective of the application is to check and demonstrate the automated RSMed-AEM program. The progress of the application is reported through the methodology steps (discussed in Chapter Three): the AEM Setup, Model Postulation, Experimental Design Selection, Data Collection, Regression Application, Coefficient Decoding, Estimated Model Completion, and Estimated Model Validation.

The AEM Setup Step

The AEM setup is twofold: formulating and recording. Formulating embodies understanding and establishing the AEM scenario; recording involves preparing and establishing the inputs for AEM executions: an AEM input file.

AEM Setup Formulation. The AEM setup formulation is composed of four parts: choosing the variables of interest, establishing a target base, selecting targeting strategies and priorities, and picking the dependent variable(s) for the measures of effectiveness.

Variables of Interest. A hypothetical eighteen variable problem is used with the application of the automated RSMed-AEM program. Each variable represents a weapon used by the U.S. to strike the Soviet Union as controlled by the AEM. Each weapon is listed in Figure 4.1 along with its range of values, carrier type, and the number of warheads per carrier. (Figure 4.1 shows eighteen variables but only lists

Number	ABM Variable	Range	Carrier	Warheads/Carrier
1	MMII	0 - 450	ICBM	1
2	MMIII	0 - 250	ICBM	3
	MMIIIA	100 - 300	ICBM	3
3	PKeeper(MX)	0 - 100	ICBM	10
4	MICBM	0 - 200	ICBM	1
5	PosC3	176 - 304	SLBM	10
6	PosC4	192 - 192	SLBM	8
	TriC4	48 - 192	SLBM	8
7	TriD5	0 - 336	SLBM	8
8	B52GRV	0 - 156	Aircraft	4
9	B52SRM	0 - 156	Aircraft	4
10	B52CMC	40 - 156	Aircraft	12
11	FB111GRV	0 - 60	Aircraft	2
12	FB111SRM	0 - 60	Aircraft	4
13	B1BGRV	30 - 100	Aircraft	4
14	B1BSRM	30 - 100	Aircraft	4
15	B1BALCM	30 - 100	Aircraft	8
16	ATB	0 - 124	Aircraft	12

Fig 4.1 The Initial Eighteen ABM Variables

sixteen numbers. Four of the weapons will be combined to form two weapons. Therefore, eighteen weapons will exist in sixteen variables. More information is given under the Experimental Design Selection Step section of this chapter.)

<u>The Target Base</u>. The target base is developed from hypothetical data used with an AFIT class assignment. The hypothetical data listed low and high target projections over a ten year period starting with year 1985. The high projection of year 1990 is taken as the target base. However, after making some preliminary ABM executions with the ABM weapon variables listed in Figure 4.1, the number of targets was adjusted downward to allow the ABM to produce high damage expectancy

Number	Target	Amount
1	Civil	140
2	Local	215
3	C3I	450
4	AFBase	70
5	ICBM Silo	640
6	LCC	120
7	MICBM	150
8	NukStor	50
9	SubPorts	20
10	SS-20	400
11	Stores	430
12	Facil	520
13	Depos	550
14	Naval	130
15	Factor	1100
16	POL	1300
17	Energy	435

Fig 4.2 The AEM Target Base

calculations through the application. Figure 4.2 shows the AEM target base used, after the adjustment, in the application of the automated RSMed-AEM program. The targets are listed by name along with the number of each target shown immediately to the right of its names.

Figure 4.3 shows the same targets, only grouped into categories. The purpose for the categorization is to later allow the usance of strategies and prioritization of goals by category rather than by specific target. Through the categorization of the targets, the five objectives of the AEM problem (described in Chapter One) are defined.

Selecting Strategies and Priorities. Targeting strategies and priorities help create a realistic arsenal exchange environment. Therefore, four strategies and six priorities are used. The targeting

Number	Target Category	Targets
1	Leadership/Command, Control, Communications, and Intelligence (LC3I)	1 thru 3
2	Strategic Nuclear	4 thru 10
3	Conventional	11 thru 14
4	Industrial/Economic	15 thru 17
5	Time Urgent	3 thru 6 and 9

Fig 4.3 AEM Target Categories
(Reference Figure 4.2 for target number definitions)

strategies and priorities are listed in Figure 4.4. The priorities are "soft" priorities instead of "hard" priorities. "Soft" priorities allow the AEM to do as well on a particular priority as possible before moving on to the next lower priority. "Hard" priorities do not allow the targeting of lower priority objectives until the higher priority objectives are fulfilled.

<u>The Dependent Variable(s)</u>. The dependent variable is damage expectancy (DE): what is the expected damage to any and all targets given the AEM weapons and targeting strategies and priorities? For the RSMed-AEM application, five dependent variables exist. Except for the Time Urgent targets, each target category listed in Figure 4.3 depicts a dependent variable. The fifth dependent variable represents the damage expectancy for all targets. (It is the estimated equation for the fifth objective that is later used to show the predictability and informational analysis advantages of Smith's RSM variant.) Each dependent variable and its associated target category is shown in Figure 4.5.

Targeting Strategy One:

Limit the maximum number of RVs from any carrier allowed to impact on a target to three.

Targeting Strategy Two:

Exclude the probability of arrival from calculations of impacts limit.

Targeting Strategy Three:

Set minimum damage expectancy (DE) limits for each target:

Target Number	Minimum DE (percentage)
1 thru 10	80
11 thru 14	70
15 thru 17	60

Targeting Strategy Four:

Specify which weapons are allowed to attack a target:

Target Number	Weapons Allowed To Attack
1, 2, 8, and 11 thru 17	1 thru 16
3 thru 6 and 9	1 thru 7
7 and 10	13, 14, and 16

Targeting Priorities:

Priority	Cover Targets By Category or Number
1	Time Urgent
2	1 and 2
3	7, 8, and 10
4	Conventional
5	Industrial/Economic
6	All

Fig 4.4 AEM Targeting Strategies and Priorities (Reference Figures 4.1 thru 4.3 for the definition of each weapon and target number/category)

Dependent Variable	Target Category
Y1	LC3I
Y2	Strategic Nuclear
Y3	Conventional
Y4	Industrial/Population
Y5	All

Fig 4.5 The AEM Dependent Variables

Recording The AEM Setup: The AEM Input File. Once the AEM setup is formulated, it is recorded to be used later for executing the AEM. Recording of the AEM formulation is done through the AEM input file: IAEM1.DAT. The IAEM1.DAT file contains the AEM code that represents each element of the AEM formulation. An example of the IAEM1.DAT file from one of the execution runs is shown in Appendix E. The execution example shows the four categories of the AEM setup: the variables of interest (weapons), a target base, the targeting strategies and priorities, and the dependent variables.

The Model Postulation Step

The second part of the RSMed-AEM methodology is postulation of a model that can represent the AEM output given an AEM setup as described in the AEM Setup Step section of this chapter. For this thesis, and as discussed in Chapter Three, model postulation will always be a second-order polynomial equation (equation 2.2). However, this need not be so. If the RSMed-AEM program allowed the user the flexibility of choosing an equation from among many, this step would have to be considered. The

user would select an equation to best represent the AEM output given an AEM setup scenario (discussed in The AEM Setup Step section of this chapter). Consequently, The Model Postulation step of the RSMed-AEM methodology is included within this application write-up for completeness and as a reminder to the user that the postulated model is dependent on the AEM Setup scenario. If the user does not believe that equation 2.2 will accurately represent a particular AEM setup scenario, then this RSMed-AEM program should not be used since the RSMed-AEM program is designed to "fit" a second-order polynomial equation.

The Experimental Design Selection Step

As with the AEM setup step, the Experimental Design Selection step is also twofold: formulation and recording. The formulation part involves selecting an experimental design and then, if necessary, manipulating the AEM variables into the required number of design variables needed to fit the chosen experimental design. Recording involves using the RSMed-AEM program to interactively enter all information pertaining to the design variables.

Experimental Design Selection Formulation. When choosing the AEM variables, thought must be given to which experimental design will be used since the selection of experimental designs are limited. As previously mentioned, eighteen AEM variables are used for the RSMed-AEM program application, but the largest experimental design built into the RSMed-AEM program is one for sixteen variables. Therefore, reduction techniques, discussed in Chapter Three, are applied to reduce the number of AEM variables.

Combination	AEM Variables	Design Variable
1	B52GRV B52SRM	B52PEN
2	FB111GRV FB111SRM	FB111PEN
3	B1BGRV B1BSRM B1BALCM	B1B

Fig 4.6 The Design Variables Established From Using The Variable Control Reduction Technique

The first variable-reduction technique used is the AEM variable control technique: how many design variables, if any, can control more than one AEM variable? A closer examination of Figure 4.1 shows that seven of the eighteen AEM variables can be controlled by three design variables. A B-52 can carry SRAMs and gravity bombs at the same time, thus enabling the combination of the B-52SRM and B-52GRV AEM variables into a B-52PEN design variable. Similar combinations can be made of the FB111 AEM variables and the B1B AEM variables. Figure 4.6 shows the design variables produced from the three different AEM variable combinations. (Note: All three B1B AEM variables are combined into one single B1B design variable, whereas the three B52 AEM variables require two design variables: one for the B52CMC and one for the combination of the B52GRV and B52SRM. This stems from assuming that a B1B can carry both gravity bombs and cruise missiles at the same time and that a B-52 cannot.)

Number	Design Variable	Range	Carrier	Warheads/Carrier
1	MMII	0 - 450	ICBM	1
2	MMIII	0 - 250	ICBM	3
	MMIIIA	100 - 300	ICBM	3
3	PKeeper(MX)	0 - 100	ICBM	10
4	MICBM	0 - 200	ICBM	1
5	PosC3	176 - 304	SLBM	10
6	PosC4	192 - 192	SLBM	8
	TriC4	48 - 192	SLBM	8
7	TriD5	0 - 336	SLBM	8
8	B52PEN	0 - 156	Aircraft	8
9	B52CMC	40 - 156	Aircraft	12
10	FB111PEN	0 - 60	Aircraft	6
11	B1B	30 - 100	Aircraft	16
12	ATB	0 - 124	Aircraft	12

Fig 4.7A Inclusion of the Control Variables

After applying the control-variable reduction technique, eighteen weapons are reduced to fourteen design variables. The results are shown in Figure 4.7A. (Likened to Figure 4.1, fourteen variables are shown but only twelve are listed. The explanation is forthcoming.) No Box-Behnken experimental design exists, however, for fourteen design variables. The closest design is either the twelve or sixteen variable experimental designs. Since a sixteen-variable design requires almost twice as many runs as a twelve-variable design, the twelve-variable design is selected.

To reduce fourteen variables to twelve, a second variable-reduction technique is employed: the combination of similar variables. For the application, the manipulation involves combining sets of similar weapons. Looking again at Figure 4.1, eighteen variables are shown but

Number	ABM Variable	Range	Carrier	Warheads/Carrier
1	MMII	0 - 450	ICBM	1
2	MMIII	100 - 550	ICBM	3
3	PKeeper(MX)	0 - 100	ICBM	10
4	MICBM	0 - 200	ICBM	1
5	PosC3	176 - 304	SLBM	10
6	PosTriC4	240 - 384	SLBM	8
7	TriD5	0 - 336	SLBM	8
8	B52GRV	0 - 156	Aircraft	4
9	B52SRM	0 - 156	Aircraft	4
10	B52CMC	40 - 156	Aircraft	12
11	FB111GRV	0 - 60	Aircraft	2
12	FB111SRM	0 - 60	Aircraft	4
13	B1BGRV	30 - 100	Aircraft	4
14	B1BSRM	30 - 100	Aircraft	4
15	B1BALCM	30 - 100	Aircraft	8
16	ATB	0 - 124	Aircraft	12

Fig 4.7B The Combination of Similar Variables
(Refer to Figure 4.1)

only sixteen are numbered. This was done in anticipation of using the similar variables, variable-reduction technique. From Figure 4.1, two sets of similar variables exist: the combination of the MMIIIs with the MMIIIAs, and the combination of the Poseidon and Trident C4 SLBMs. The weapons in both sets are very similar and therefore allow the variable reducing merger. Figure 4.7B shows the change in Figure 4.1 after the similar variables are combined.

After applying the combination of similar variables, variable-reduction technique, the fourteen design variables are reduced to twelve. The assumption for using the similar variables, variable-reduction technique is that informational analysis is not important for those four individual weapons that are merged. If it is, then the

Number	Design Variable	Range	Carrier	Warheads/Carrier
1	MMII	0 - 450	ICBM	1
2	MMIII	100 - 550	ICBM	3
3	PKeeper(MX)	0 - 100	ICBM	10
4	MICBM	0 - 200	ICBM	1
5	PosC3	176 - 304	SLBM	10
6	PosTriC4	240 - 384	SLBM	8
7	TriD5	0 - 336	SLBM	8
8	B52PEN	0 - 156	Aircraft	8
9	B52CMC	40 - 156	Aircraft	12
10	FB111PEN	0 - 60	Aircraft	6
11	B1B	30 - 100	Aircraft	16
12	ATB	0 - 124	Aircraft	12

Fig 4.8 The Finalized Twelve Design Variables

mergers would not occur; rather two dummy design variables would be used to increase the number of design variables from fourteen to sixteen since there is a Box-Behnken design for sixteen design variables. But, with the assumption, the reduced set of variables (weapons), from eighteen AEM variables to twelve design variables, is listed in Figure 4.8, along with the updated range, the carrier type, and the number of warheads per carrier.

Recording the Experimental Design Selection. The design variable information contained in Figure 4.8 is now entered into the RSMed-AEM program. The RSMed-AEM program is started (refer to Appendix C), and subroutine INPTDATA is chosen (selection I) once the main menu appears via the CRT terminal (Figure 4.9). After the INPTDATA subroutine is started, the information on the twelve design variables is entered and saved in a file entitled VARIBLES.DAT (Appendix F). Three other files

```
****************************** MAIN MENU ******************************

                    (I)  Input Data

                    (U)  Build Uncoded Design

                    (C)  Build Coded Design

                    (R)  Regression

                    (D)  Decode

                    (X)  Execute Equations

                    (E)  Exit Program


 Please enter a letter:
```

Fig 4.9 Automated RSM Program's Main Menu

are created by the INPTDATA subroutine to be used later as input with other main-menu subroutines: the CODEDSGN.IN file (shown in Appendix G) for the CODEDSGN subroutine, the DECODE.IN file (shown in Appendix H) for the DECODE subroutine, and the RGRSSION.IN file (Appendix I) for the RGRSSION subroutine.

<u>Data Collection Step</u>.

After using subroutine INPTDATA to enter the design variable information and saving that information in the file VARIBLES.DAT, the main menu reappears, and subroutine CODEDSGN is chosen (selection U).

Subroutine CODEDSGN does not interact with the user; rather, subroutine CODEDSGN just creates a file. When selection U is chosen, the file AEM.DAT (Appendix J) is created for later use with Tool Two (refer to Chapter 3). After the AEM.DAT file is created, the main menu reappears and selection E is chosen; the RSMed-AEM program is exited.

With the AEM input file (IAEM1.DAT) and the AEM experimental design file (AEM.DAT) created, Tool Two (Appendix D) is used with the AEM to control and amass the data collection. Tool Two automatically executes the AEM 193 times. For each execution, an AEM output file (OAEM1.DAT) is created (Appendix K). With each execution, input from the AEM.DAT file is collected, along with the dependent variable values from the OAEM1.DAT file (Appendix K), and stored in a file entitled STORAGE.DAT (Appendix L).

Regression Step.

So far, an AEM scenario has been coded into an IAEM1.DAT file; an AEM experimental design file entitled AEM.DAT has been created; and Tool Two has used those two files to control the AEM executions while collecting and storing data generated by each execution in a file called STORAGE.DAT. To execute the regression package on the RSMed-AEM program, the data in file AEM.DAT has to be coded according to equation 2.3 and the data in file STORAGE.DAT has to be modified to only show the responses of the dependent variables.

The STORAGE.DAT file is first modified by manually stripping all the lines containing the values for the AEM weapons. The modification is accomplished by deleting every other line starting with line one. The modified file, RESPNSES.IN, is shown in Appendix M.

To code the AEM.DAT file, the RSMed-AEM program is restarted. Once the main menu appears, the INPTDATA subroutine is first chosen (selection I) to re-initialize the design variable information. Re-initialization is done by reading the VARIBLES.DAT file again. (Recall the VARIBLES.DAT file saved the design variable characteristics that were previously interactively entered.) After re-initialization is accomplished, the main menu reappears. The CODEDSGN subroutine is again chosen, but this time selection C is picked. As noted earlier, the CODEDSGN subroutine does not interact with the user; it just creates a file. With selection C chosen, the CODEDSGN subroutine uses the CODEDSGN.IN file to create the DSGNMTRX.IN file (Appendix N). Once the DSGNMTRX.IN file is created, the main menu reappears.

With the creation of the three required input files now complete (the third input, the RGRSSION.IN file, was created earlier by the INPTDATA subroutine), the RGRSSION subroutine is chosen from the main menu (selection R). All five objectives (the dependent variables listed in Figure 4.5) are processed through the RGRSSION subroutine. The results, except for coefficient estimates, from the RGRSSION subroutine are written to the REGRESS.DAT file (Appendix O) to be printed later. Once the regression process is complete, the RGRSSION subroutine writes the estimated coefficients to the CODEDBTS.IN file (Appendix P) and then releases control back to the main menu.

To verify that the RGRSSION subroutine gives correct answers, the RGRSSION output for the fifth objective is compared to regression output from the SAS package using the same inputs. SAS is a widely accepted statistical package. SAS's input code is shown in Appendix Q. SAS's output, in Appendix R, contains the same values for SSR, SSE, SSTO, R^2,

and residual statistics along with the coded parameter estimates and individual sum-of-squares (Type I Sum-Of-Squares) as does the RGRSSION output for the fifth objective shown in Appendices O and P.

The Decode Coefficients Step

The RGRSSION subroutine has now estimated the coefficients to the postulated model (equation 2.2). However, a coded design was used for the estimation. Once the main menu reappears, the DECODE subroutine is chosen (selection D). The DECODE.IN file (created by the INPTDATA subroutine) and the CODEDBTS.IN file (created by the RGRSSION subroutine) are used as input by the DECODE subroutine. The coded parameter estimates in the CODEDBTS.IN file are decoded and stored in the DECODE.OUT and XCUTEQNS.IN files (Appendices T and S). The main menu reappears when the parameter decoding is complete.

The Estimated Model Completion Step

With the parameter estimates decoded, the estimated model can now be completed. The decoded parameter estimates listed in the XCUTEQNS.IN file are joined with their design variables. The process of joining the parameter estimates with their respective design variables appears complicated at first, but is easy to accomplish after demonstrating the process.

Figure 4.8 and Appendix T are used for the demonstration. Completion of the postulated model (equation) for the fifth objective will be demonstrated. From Appendix T, the parameter estimates for the fifth ^bjective are located in the last 92 lines. The first line informs the user that the following parameter estimates belong to the fifth objective.

The joining process starts with the first N + 1 parameter estimates where N is the number of design variables (twelve in this example). The INTERCEPT value is listed first followed by N linear parameter estimates. The N linear parameter estimates are joined with their respective design variables in the order given by Figure 4.8 (i.e., the next parameter estimate after the INTERCEPT term is joined with the MMII design variable; the following parameter estimate is joined with the MMIII design variable; and et al).

The first N + 1 parameter estimates are joined with the INTERCEPT and the linear design variable terms. The remaining N! parameter estimates are joined with the squared and cross design variable terms. The procedure for joining the remaining N! parameter estimates is to successively descend the design variable list (Figure 4.8), one design variable at a time, matching the next parameter estimate with the squared term of the design variable first and then matching the following parameter estimates with the cross terms that result by multiplying that design variable with the remaining design variables. Once the squared and cross terms for the first design variable are joined (MMII in this example), descend the design variable list (Figure 4.8) to the next design variable (MMIII) and repeat the procedure, but do not include any cross terms that have already been joined previously. The completion of the postulated model for the fifth objective (All Targets) is shown in Appendix U.

The Estimated Model Validation Step

Validation is accomplished by comparing the responses from the estimated models with the responses from the AEM for the same set of

Set	MMII	MMIII	PKEEP	MICBM	POSC3	POSTRIC4	TRID5	B52GRV	B52SRM	B52CMC	FB111GRV	FB111SRM	B1BGRV	B1BSRM	B1BALCM	ATB
	AEM Variables															
1	350	375	20	120	200	302	188	58	58	58	40	40	75	75	75	42
2	250	300	88	188	277	262	200	99	99	128	56	56	55	55	55	22
3	440	175	20	133	199	266	222	44	44	50	44	44	50	50	50	32
4	277	189	12	166	271	367	75	33	33	144	42	42	45	45	45	112
5	369	533	93	104	219	380	61	150	150	77	57	57	35	35	35	10
6	140	402	83	35	208	284	311	17	17	105	47	47	85	85	85	99
7	77	110	45	10	190	250	10	5	5	42	33	33	95	95	95	102
8	190	410	38	150	290	333	48	137	137	115	50	50	33	33	33	80
9	400	500	25	25	185	370	175	125	125	70	38	38	40	40	40	88
10	175	173	76	80	300	277	99	110	110	133	35	35	48	48	48	120
11	355	370	25	115	205	297	193	53	53	63	5	5	80	80	80	37
12	245	305	83	193	272	267	195	104	104	123	60	60	50	50	50	27
13	445	170	25	128	204	261	227	39	39	55	9	9	55	55	55	27
14	272	194	7	171	266	362	80	28	28	149	7	7	50	50	50	107
15	374	528	98	99	224	375	66	145	145	82	22	22	40	40	40	5
16	135	407	78	40	203	289	306	22	22	100	12	12	80	80	80	104
17	82	105	50	5	195	245	15	0	0	47	0	0	100	100	100	97
18	185	415	33	155	285	338	43	142	142	110	25	25	30	30	30	85
19	405	495	30	20	190	365	180	120	120	75	3	3	45	45	45	83
20	170	178	71	85	295	282	94	115	115	128	10	10	43	43	43	124

Fig 4.10 Validation Test Points

values assigned the design variables (weapons). For the application, only the estimated model for the fifth objective is validated. To validate the fifth objective's estimated model, twenty randomly chosen points (values for the design variables), other than the experimental design points, are chosen. The design variables and the twenty randomly chosen test points are shown in Figure 4.10.

Set	First Objective	Second Objective	Third Objective	Fourth Objective	Fifth Objective
1	87.54	83.63	77.80	41.17	64.77
2	83.77	73.63	79.81	78.25	78.29
3	85.68	75.08	77.56	32.13	58.83
4	87.13	61.84	77.82	75.80	74.64
5	88.08	62.48	75.27	74.54	73.74
6	84.96	89.91	81.60	82.26	84.07
7	85.86	43.45	75.04	50.67	59.24
8	88.40	67.53	74.97	74.27	74.68
9	86.60	85.87	75.54	68.35	76.06
10	86.73	81.91	81.57	73.10	78.69
11	87.22	83.17	77.59	43.88	65.73
12	84.66	74.12	79.29	77.30	77.98
13	85.08	74.56	77.47	34.99	59.83
14	87.16	61.43	76.69	75.21	74.03
15	88.40	64.18	75.61	73.62	73.84
16	84.07	89.88	82.51	80.45	83.42
17	89.12	44.87	75.55	47.29	58.64
18	88.33	66.25	75.74	72.84	73.98
19	86.81	88.13	76.06	64.48	75.07
20	86.76	79.75	80.23	73.02	77.87

Fig 4.11 AEM Responses To The Validation Test Points

The twenty points are first used to execute the AEM. The AEM.DAT file is manually setup with the data from Figure 4.10. Tool Two is again used to control the AEM executions and to collect the output (the response) for the fifth objective and store it in the STORAGE.DAT file. Only the fifth objective is validated, but Figure 4.11 shows the responses for each of the five objectives for each AEM execution.

After the AEM executions, the twenty points are used to execute the estimated model for the fifth objective only (Appendix U). The RSMed-AEM program is restarted, and the Regression and Decode Coefficients Steps (refer to those sections in this chapter) are repeated. After the

Execution	Fifth Objective's Response
1	67.93
2	79.24
3	58.25
4	75.75
5	75.67
6	84.43
7	52.29
8	74.93
9	75.30
10	79.54
11	67.72
12	78.54
13	57.90
14	74.41
15	75.56
16	83.61
17	52.40
18	73.82
19	74.44
20	78.39

Fig 4.12 Estimated Model Responses To The Validation Test Points

DECODE subroutine is finished and the main menu reappears, the XCUTEQNS subroutine is picked (selection X). The fifth objective is chosen for execution. By line, the data from Figure 4.10 is interactively entered into the XCUTEQNS subroutine which then executes the estimated equation for the fifth objective and displays the response for each execution via the CRT. The responses are shown in Figure 4.12.

After the responses from the AEM and the estimated model executions for the fifth objective are collected, the residuals are calculated along with the residuals' mean and standard deviation (Figure 4.13). The residuals' mean (0.164) is quite small but a little

Execution	AEM Response	Postulated Model Response	Residual
1	64.77	67.93	- 3.16
2	78.29	79.24	- 0.95
3	58.83	58.25	+ 0.58
4	74.64	75.75	- 1.11
5	73.74	75.67	- 1.93
6	84.07	84.43	- 0.36
7	59.24	52.29	+ 6.95
8	74.68	74.93	- 0.25
9	76.06	75.30	+ 0.76
10	78.69	79.54	- 0.85
11	65.73	67.72	- 1.99
12	77.98	78.54	- 0.56
13	59.83	57.90	+ 1.93
14	74.03	74.41	- 0.38
15	73.84	75.56	- 1.72
16	83.42	83.61	- 0.19
17	58.64	52.40	+ 6.24
18	73.98	73.82	+ 0.16
19	75.07	74.44	+ 0.63
20	77.87	78.39	- 0.52
	Residuals' Mean		+ 0.164
	Residuals' Standard Deviation		+/- 2.479

Fig 4.13 Statistical Table For The Validation Test

misleading since three residuals (-3.16, +6.95, and +6.24) are above the residuals' standard deviation (+/- 2.479). Figure 4.10 shows that the two largest residuals are from the test data point sets seven and seventeen. Those two sets had considerably fewer weapons than did the other test data point sets. Also, a comparison of the residuals for those three test points with the residuals for the experimental design points that are close to the test points (Appendix O) might further explain why those three test points exceed the standard deviation.

Conclusion

This chapter describes the application given the RSMed-AEM methodology. The description of the application is explained through the steps of the automated RSMed-AEM methodology, explained in Chapter Three. Next, predictability and informational analysis are applied to the estimated models (full and reduced) and reported in Chapter Five.

Chapter Five: Analysis

Introduction

This chapter shows the advantages of applying RSM to a deterministic model (the AEM). The advantages are predictability and informational analysis. Only the fifth objective, the damage expected from all targets, is used to show the advantages.

Predictability

Once the coefficients of a postulated model have been estimated, the now estimated model takes far less time in calculating the response (damage expectancy) than does the deterministic model. Calculating the response in less time is the predictability advantage that a estimated model has over the original deterministic model.

Predictability has already been demonstrated in the Estimated Model Validation step from Chapter Four. After the postulated model was "fitted" with the decoded parameter estimates (from the DECODE subroutine), the XCUTEQNS subroutine was used to calculate damage expectancy responses given the validation test data of Figure 4.10. Once one of the validation test points was entered, via the terminal, the response was calculated and printed to the screen in less than a second. It took the AEM between one and two minutes to calculate a response given the same validation test point.

The time difference between the estimated model's execution and the AEM's execution does not seem very significant, but remember that the AEM execution time is dependent on input strategies and priorities. If more complex targeting strategies and priorities are used, the AEM would

require more execution time. In fact, recall from Chapter One that the AF/XOXR-MAA uses targeting strategies and priorities that require approximately three hours for one AEM execution. If a postulated model was constructed for a common AF/XOXR-MAA targeting strategy and priority problem, the estimated model's advantage of predictability would be more easily recognized.

Using an estimated model's predictability advantage is not without penalty. From the validation test results in Chapter Four, a trade-off exists: speed versus accuracy. When the estimated model was used to calculate a response, some accuracy was lost (Figure 4.13). So, the advantage of predictability is available only if the user can accept the inaccuracy.

Informational Analysis

The estimated model is extremely useful for informational analysis. However, working with the full model of Appendix U would be frustrating and time consuming. Therefore, if the full estimated model can be reduced (delete terms that do not contribute much to the response), the resulting reduced estimated model could be used more effectively and efficiently for informational analysis.

Obtain Reduced Estimated Model. The RGRSSION subroutine of the RSMed-AEM program is used to reduce the full estimated model. However, before the RGRSSION subroutine is used, preliminary files have to be created. So, upon entering the program and after the main menu appears, first execute the INPTDATA subroutine (selection I). The design variable data used in building the full estimated model should still be on file (VARIBLES.DAT). Therefore, after selecting the INPTDATA

subroutine, read the data from the VARIBLES.DAT (Appendix F) file and verify the data is still valid. After verification, the INPTDATA subroutine will create some files (refer to Appendix C) and release control back to the main menu. Next, execute the CODEDSGN subroutine (selection C). The CODEDSGN subroutine also creates a file (refer to Appendix C) and releases control back to the main menu.

After the preliminary steps are complete, the RGRSSION subroutine is invoked (selection R). Using the regression output for the fifth objective's full estimated model (Appendix O) as a guide, the RGRSSION subroutine allows the user to reduce the full estimated model through deletion of unnecessary terms. The REGRESS.DAT file (Appendix O) is designed to help the user select which term to delete or keep. The terms are sorted in ascending order according to their contribution to the total sum-of-squares for the complete model (SSR). After reviewing the output for the fifth objective's full estimated model (Appendix O), the last sixteen terms account for approximately ninety-five percent of the SSR. The remaining, and previous, seventy-four terms account for about five percent of the total SSR and, therefore, can be deleted without affecting the response too much.

Upon execution of the RGRSSION subroutine, the user is asked if he/she wishes to delete a term. If the user responds affirmatively, all terms, including the INTERCEPT term, are shown to the user. Each term is cross referenced with a number. For deleting the seventy-four terms, their representative numbers are entered one by one until all unwanted terms are deleted. (All user input is checked for errors. The INTERCEPT term cannot be deleted, and no term can be deleted more than once.) Appendix V shows the regression output for the reduced model.

```
Y5 = - 3.687629292556
     + 0.003225022222 * MMII
     + 0.011744444444 * MMIII
     + 0.121438027586 * PKEEP
     + 0.010754700000 * MICBM
     + 0.077041093750 * POSC3
     + 0.054804722222 * POSTRIC4
     + 0.105882105911 * TRID5
     + 0.025530897436 * B-52PEN
     + 0.115705689655 * B-52CMC
     + 0.026062666667 * FB111PEN
     + 0.094593714286 * B1B
     + 0.149263978495 * ATB
     - 0.000373920690 * PKEEP * B-52CMC
     - 0.000143694196 * POSC3 * TRID5
     - 0.000232484319 * POSTRIC4 * ATB
     - 0.000248871100 * TRID5 * B-52CMC
```

Fig 5.1: Reduced Estimated Equation For The Fifth Objective

Once the RGRSSION subroutine is complete, control is passed back to the main menu. The parameter estimates for the sixteen undeleted terms are in coded format (refer to Appendix C). To decode those parameter estimates, the DECODE subroutine is chosen from the main menu (selection D). So, after the main menu reappears, the DECODE subroutine is selected and the fifth objective's reduced model parameter estimates are decoded. The decoded output for the reduced estimated model is shown in Appendix W (DECODE.OUT file). Figure 5.1 shows the same output only in equation form.

<u>Perform Informational Analysis With The Reduced Estimated Model</u>. The reduced polynomial equation shown in Figure 5.1 is now easily used to garner information through informational analysis. Since no pure quadratic terms exist (refer to Equation 2.2 and the Experimental Design section of Chapter Three), the terms of the reduced estimated equation

are independent of each other. Therefore, because the terms are independent of each other, damage expectancy comparisons are made directly from the equation's terms. Figures 5.2 through 5.5 show some information reaped from the reduced estimated equation. But, remember that the graphs in Figures 5.2 thru 5.5 are slightly in error since the reduced estimated equation is used instead of the full. Also, the graphs in Figures 5.2 thru 5.5 are highly indicative of the ABM scenario: the targeting strategies and priorities used and the number of weapons and targets available. (Note: the analysis examples that follow were first described by Smith and Mellichamp (27).)

Figures 5.2A and 5.2B show damage expectancy increases per weapon system carrier. The values are taken directly from the coefficients of the reduced polynomial equation. Both figures show the weapon systems in descending order by their contribution per carrier to the total damage expectancy. The B-52CMC and Trid5 weapon systems are influenced by two cross terms as shown in Figure 5.1. They appear as three-dimensional boxes in Figures 5.2A and 5.2B. The PKEEP, ATB, POSC3, and POSTRIC4 weapon systems are influenced by only one cross term each. They are depicted as two-dimensional planes in Figures 5.2A and 5.2B. The remaining weapons are not influenced by any other term and, thus, are shown in Figures 5.2A and 5.2B as one-dimensional lines.

The increase in damage expectancy per weapon system shown in Figures 5.2A and 5.2B does not depend on how many weapons already exist for that particular weapon system. An additional B1B will increase damage expectancy by 0.095 percent no matter if 30 or 99 B1Bs already exist. The same applies to each weapon system. But, an increase in damage expectancy for any weapon system influenced by cross terms does

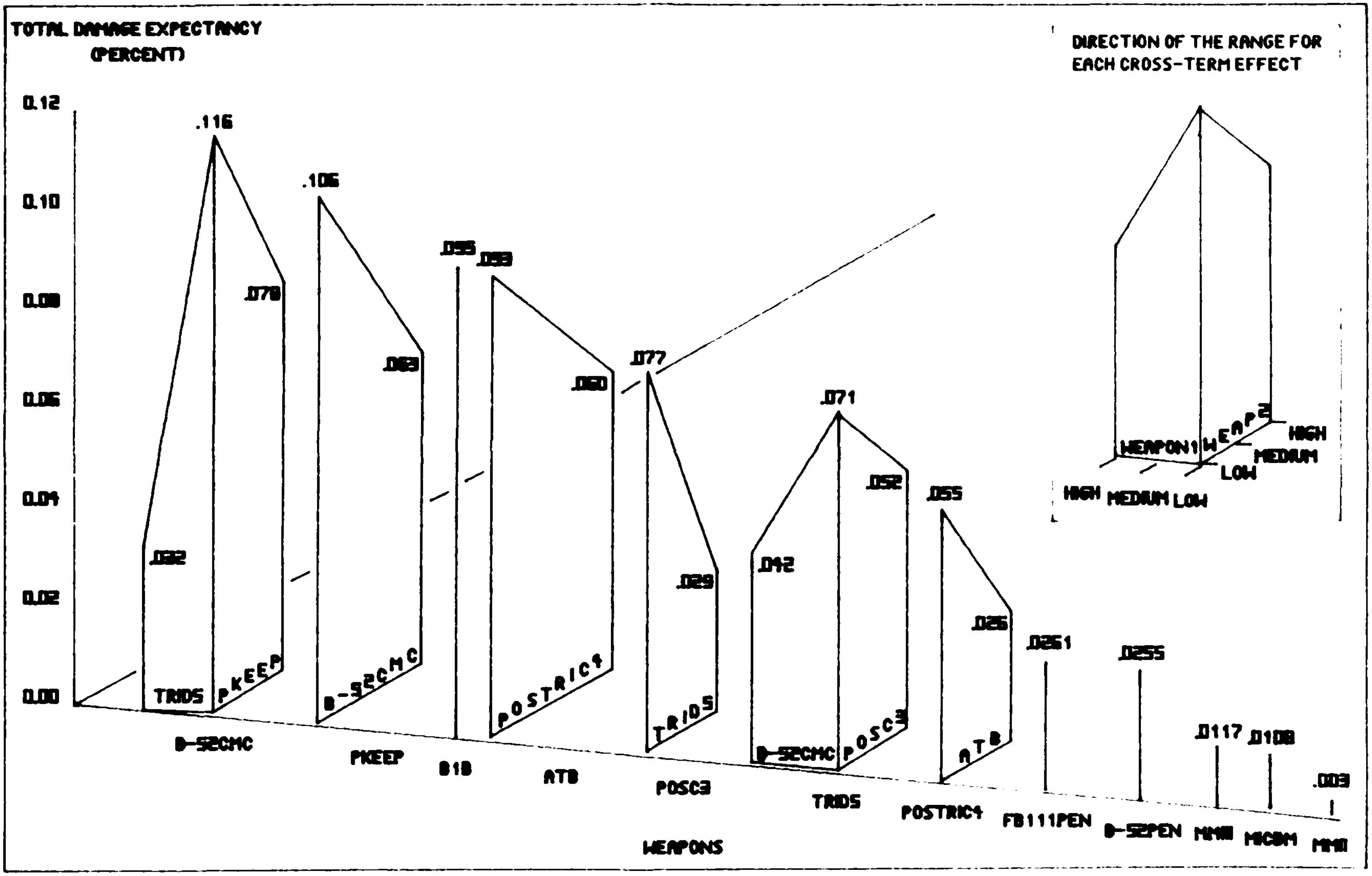
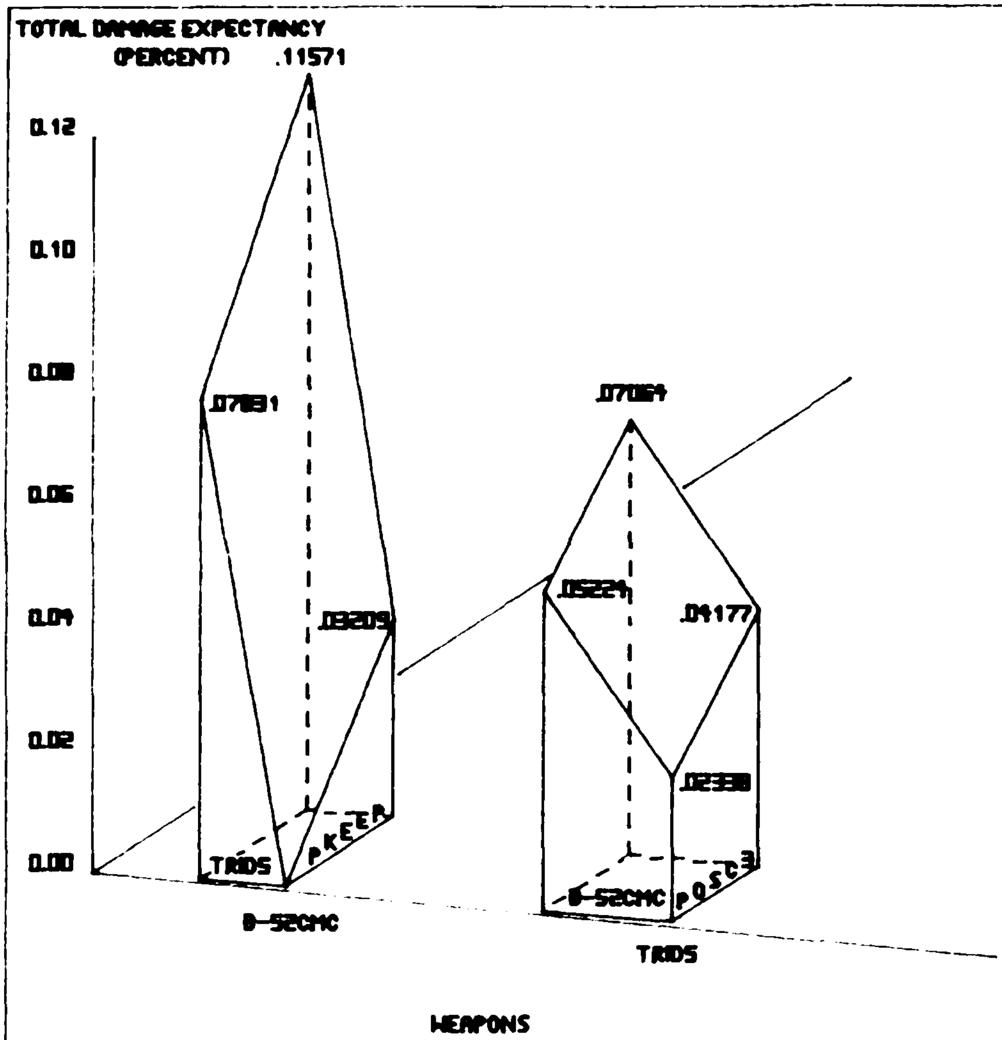


FIG 5.2A: IMPROVEMENT IN DAMAGE EXPECTANCY
PER WEAPON SYSTEM CARRIER



DAMAGE EXPECTANCY PER CARRIER FOR THE	GIVEN THE WEAPON NUMBERS OF:	
B-52CMC	PKEEP	TRIDS
0.11571	0	0
0.07390	0	168
0.03209	0	336
0.09701	50	0
0.09520	50	168
0.01339	50	336
0.07831	100	0
0.03650	100	168
0.00	100	336

DAMAGE EXPECTANCY PER CARRIER FOR THE	GIVEN THE WEAPON NUMBERS OF:	
TRIDS	POSC3	B-52CMC
0.07063	176	40
0.05620	176	98
0.04177	176	156
0.06144	240	40
0.04701	240	98
0.03257	240	156
0.05224	304	40
0.03781	304	98
0.02338	304	156

FIG 5.28: THE THREE-DIMENSIONAL VIEW OF THE DAMAGE EXPECTANCY IMPROVEMENTS PER B-52CMC AND TRIDS WEAPON SYSTEM CARRIER

depend on the number of existing cross term weapons. An example is the B-52CMC weapon. The B-52CMC provides the greatest increase in damage expectancy, but is dependent on the number of existing cross term weapons. If the B-52CMCs were increased by just one and the Trid5 and PKEEP weapons were each at their low range (Figure 4.8), the total damage expectancy would increase by 0.116 percent. If either or neither of the B-52CMC cross term weapons were at their low range, then an additional B-52CMC would not provide an increase in total damage expectancy of 0.116. Therefore, the Trid5 and PKEEP weapons have to be considered since they interact with the B-52CMC weapon.

Figure 5.2B shows the same results for the B-52CMC and Trid5 weapons as Figure 5.2A does only turned 180 degrees to reveal the three-dimensional plane. Figure 5.2B shows that adding another B-52CMC will not give any additional increase in the damage expectancy if the Trid5 and PKEEP weapons are at their high range values (Figure 4.8). This may seem a little misleading, but remember that any information received through informational analysis of the reduced estimated equation is influenced by the AEM scenario (Figures 4.1 thru 4.4). Also, using the reduced model instead of the full model provides slight error in the analysis. Besides, there is some error by just working with a postulated model that does nct identically duplicate the deterministic model it replaces (Figure 4.13). But, in lieu of the inherent slight errors caused from using the reduced estimated model, the pictorial of the B-52CMC weapon system shown in Figure 5.2B is largely indicative of the targeting strategies and priorities used and the number of weapons and targets available in the AEM scenario.

Figures 5.3A and 5.3B show the same information as Figures 5.2A and 5.2B except the damage expectancy improvements are per warhead carried by a weapon system. The weapons are again shown in descending order by damage expectancy contribution. Figure 5.3A reveals that the MICBM weapon provides the greatest increase in damage expectancy per warhead addition. In contrast, Figure 5.2A shows that the B-52CMC delivers the most damage expectancy per system carrier if its cross terms are at their low values.

The reduced estimated equation shown in figure 5.1 can also be used for weapon ratio comparisons. Because the terms of the reduced model are independent, ratios can be computed with their coefficients that show relative contributions of any weapon to all other weapons. For example, Figure 5.4 shows how many of the other weapons are required to equal the damage expectancy of one B1B. The weapons are again shown in descending order by the number of weapons needed. Again, the results of the graph are influenced by the ABM scenario and estimated equation. For example, with the B-52CMC cross terms (Trid5 and PKEEP) at their high values, Figure 5.4 shows that no amount of B-52CMCs can equivalent the damage expectancy of one B1B (represented by infinity). This, of course, does not seem correct, but, given the ABM scenario and the B-52CMC cross terms at their high values, there is no need for additional B-52CMCs (refer back to Figures 5.2B and 5.3B).

The best ratios are the B-52CMC and PKEEP weapons with their respective cross terms low. With their cross terms low, its better to use the B-52CMC and PKEEP weapons over the B1B provided all else is equal between the weapons (i.e., the targeting strategies, priorities, and scenario). But, as the B-52CMC and PKEEP cross terms increase, so

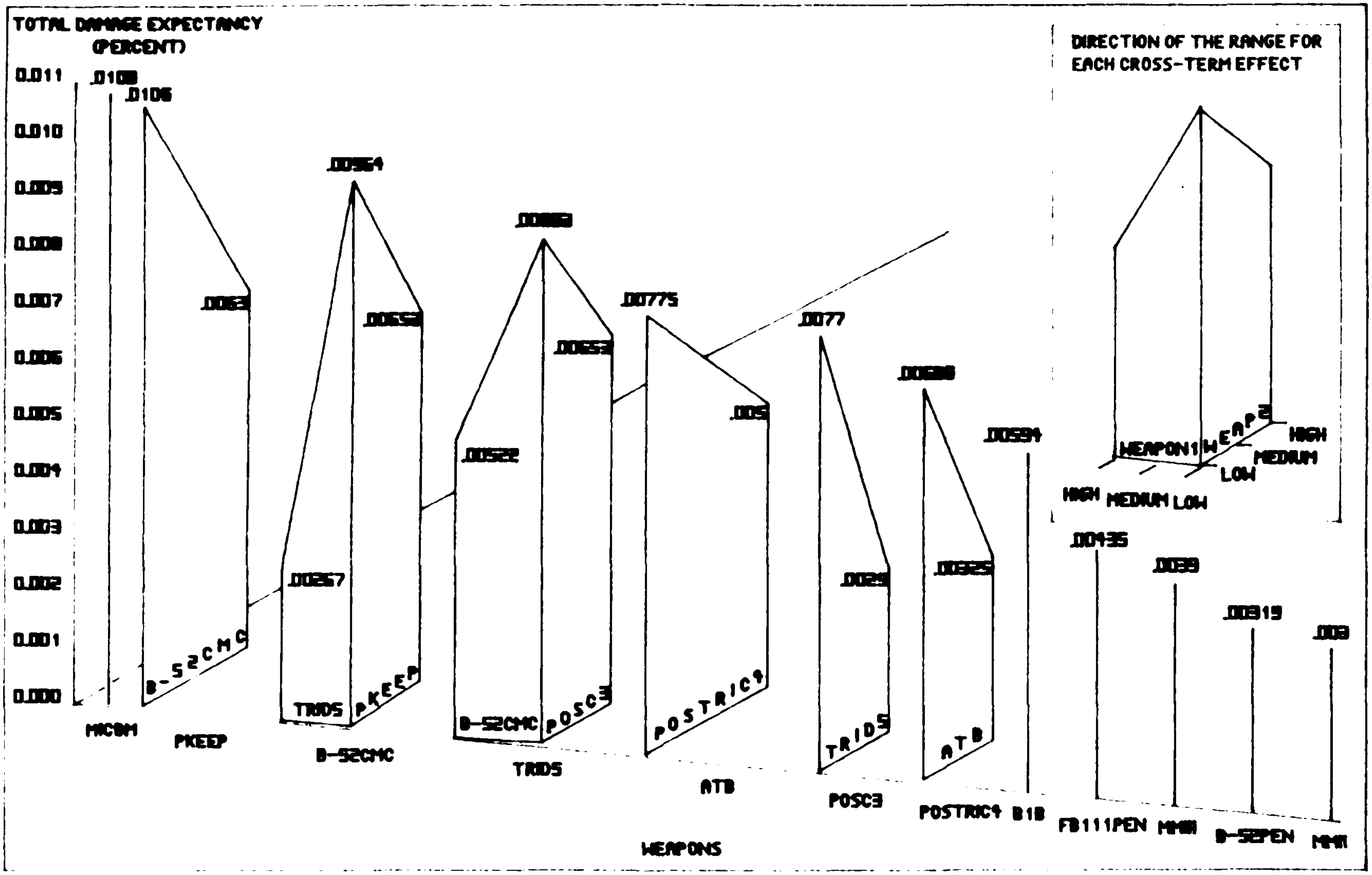
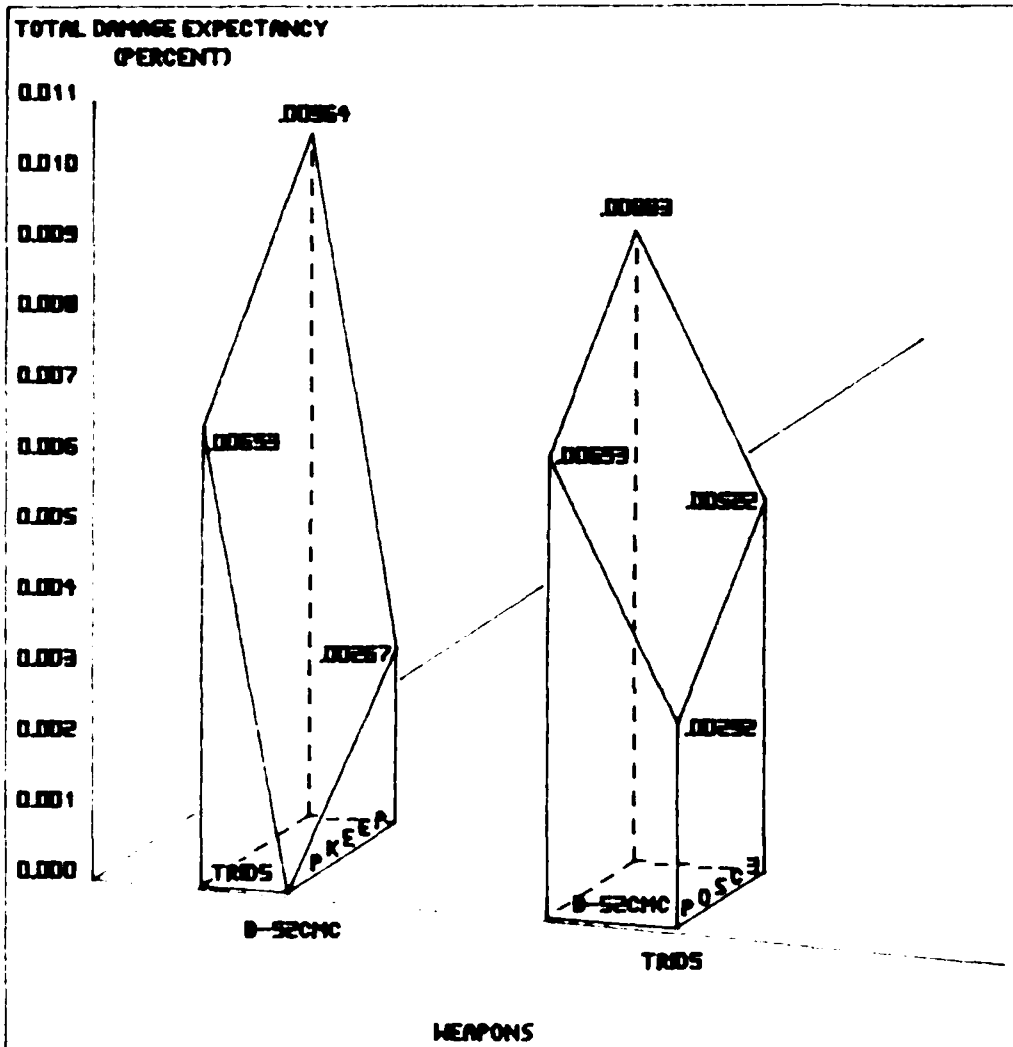


FIG 5.3A: IMPROVEMENT IN DAMAGE EXPECTANCY PER A WEAPON'S WARHEAD



DAMAGE EXPECTANCY PER WARHEAD FOR THE	GIVEN THE WEAPON NUMBERS OF:	
B-52CMC	PKEEP	TRIDS
0.00964	0	0
0.00616	0	168
0.00267	0	336
0.00808	50	0
0.00480	50	168
0.00112	50	336
0.00659	100	0
0.00304	100	168
0.00	100	336

DAMAGE EXPECTANCY PER WARHEAD FOR THE	GIVEN THE WEAPON NUMBERS OF:	
TRIDS	POSC3	B-52CMC
0.00883	176	40
0.00703	176	98
0.00522	176	156
0.00768	240	40
0.00588	240	98
0.00407	240	156
0.00653	304	40
0.00473	304	98
0.00292	304	156

FIG 5.38: THE THREE-DIMENSIONAL VIEW OF THE DAMAGE EXPECTANCY IMPROVEMENTS PER WARHEAD FOR THE B-52CMC AND TRIDS WEAPONS

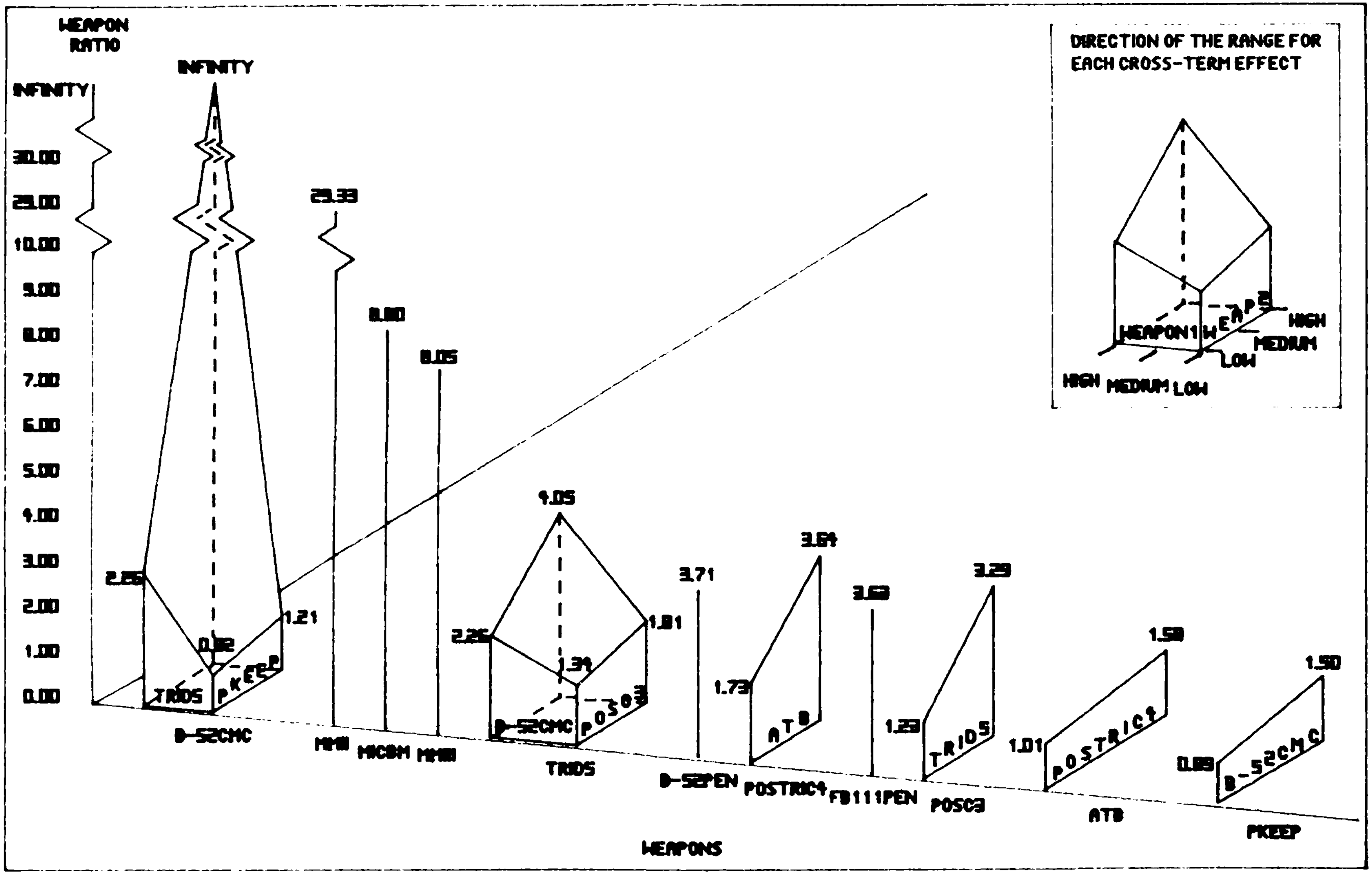


FIG 5.9: THE EQUIVALENT NUMBER OF WEAPONS REQUIRED TO EXERT THE SAME DAMAGE EXPECTANCY AS ONE B1B

do their ratios, and, soon, the B1B becomes better than the B-52CMC and PKEEP weapons, given all else remains the same.

Figures 5.5A and 5.5B show graphs containing constant damage expectancy contour lines for different combinations of the B1B, B-52PEN, and MMIII weapons. Figure 5.5A exhibits constant contour lines for a damage expectancy of 85 percent, and Figure 5.5B exhibits constant contour lines for an 87 percent damage expectancy. The remaining weapons are held constant at predetermined values.

From the graphs in Figures 5.5A and 5.5B, three observations are noticed. First, the difference between the two graphs shows that higher combinations of the three weapons are required to maintain an 87 percent contour line as opposed to an 85 percent contour line. Second, from both graphs, as the number of MMIIIs decrease, the combinations of B1Bs and B-52PENs must increase to maintain the constant damage expectancy. Third, from both graphs, for a constant MMIII value, a decrease in either the B1B or B-52PEN warrants an increase in the other to sustain the constant damage expectancy.

Figures 5.2 thru 5.5 reveals how some informational analysis can be extracted from the estimated equation. Given the AEM scenario, the figures explain the contribution of each factor towards the response. Contribution is shown by carrier and then by warhead. Also, the relative worth of one system as compared to others is displayed. Finally, graphs show what the different combinations of weapons must be to maintain constant damage expectancy contour lines.

Any insight/knowledge gained through informational analysis must be tempered with the AEM scenario, especially targeting strategies and priorities, because the AEM scenario affects the construction of the

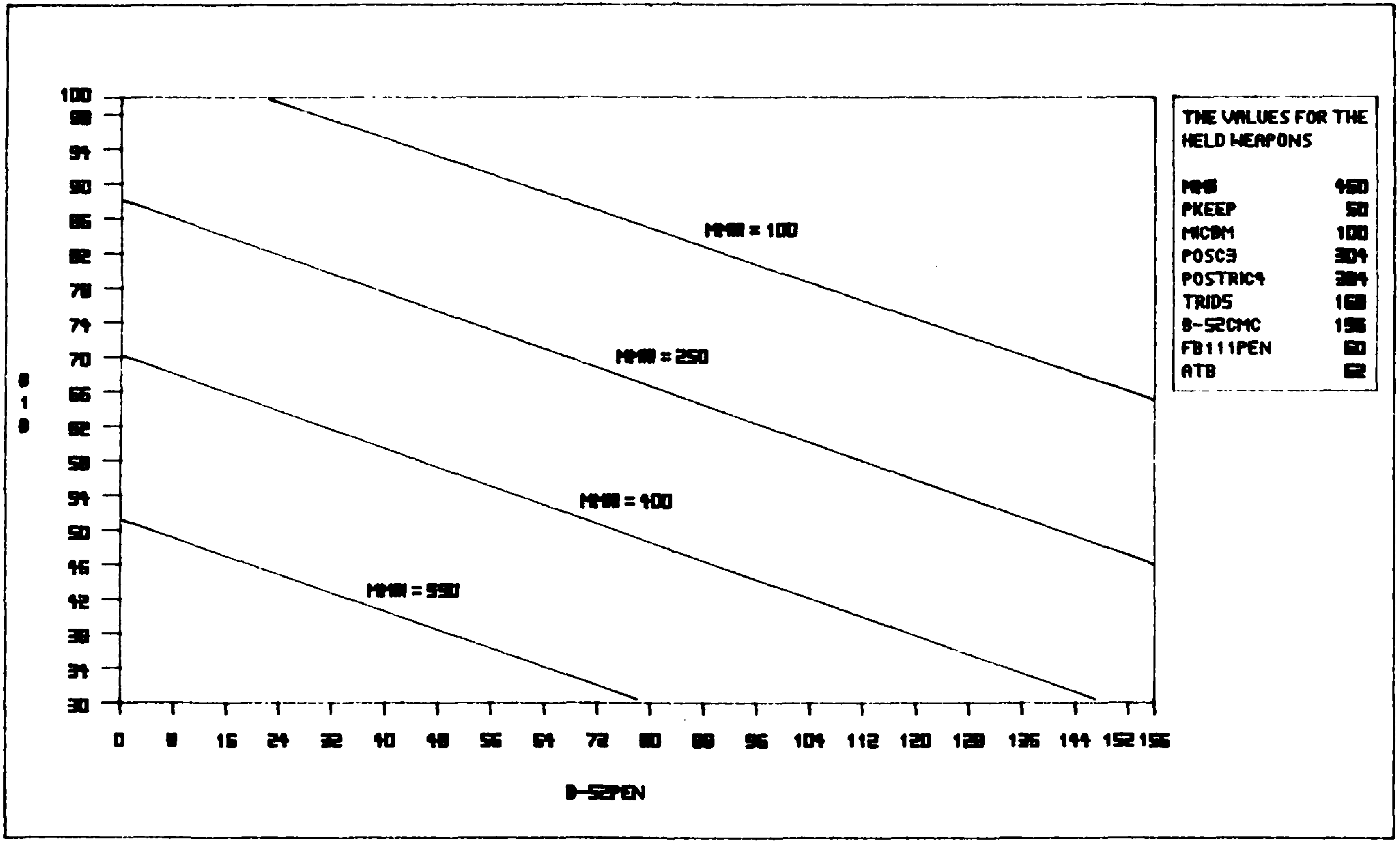


FIG 5.5A: CONSTANT RISK DE CONTOUR LINES FOR VARIOUS COMBINATIONS OF B1BS, B-52PENS, AND MMIIIS. ALL OTHER WEAPONS HELD CONSTANT.

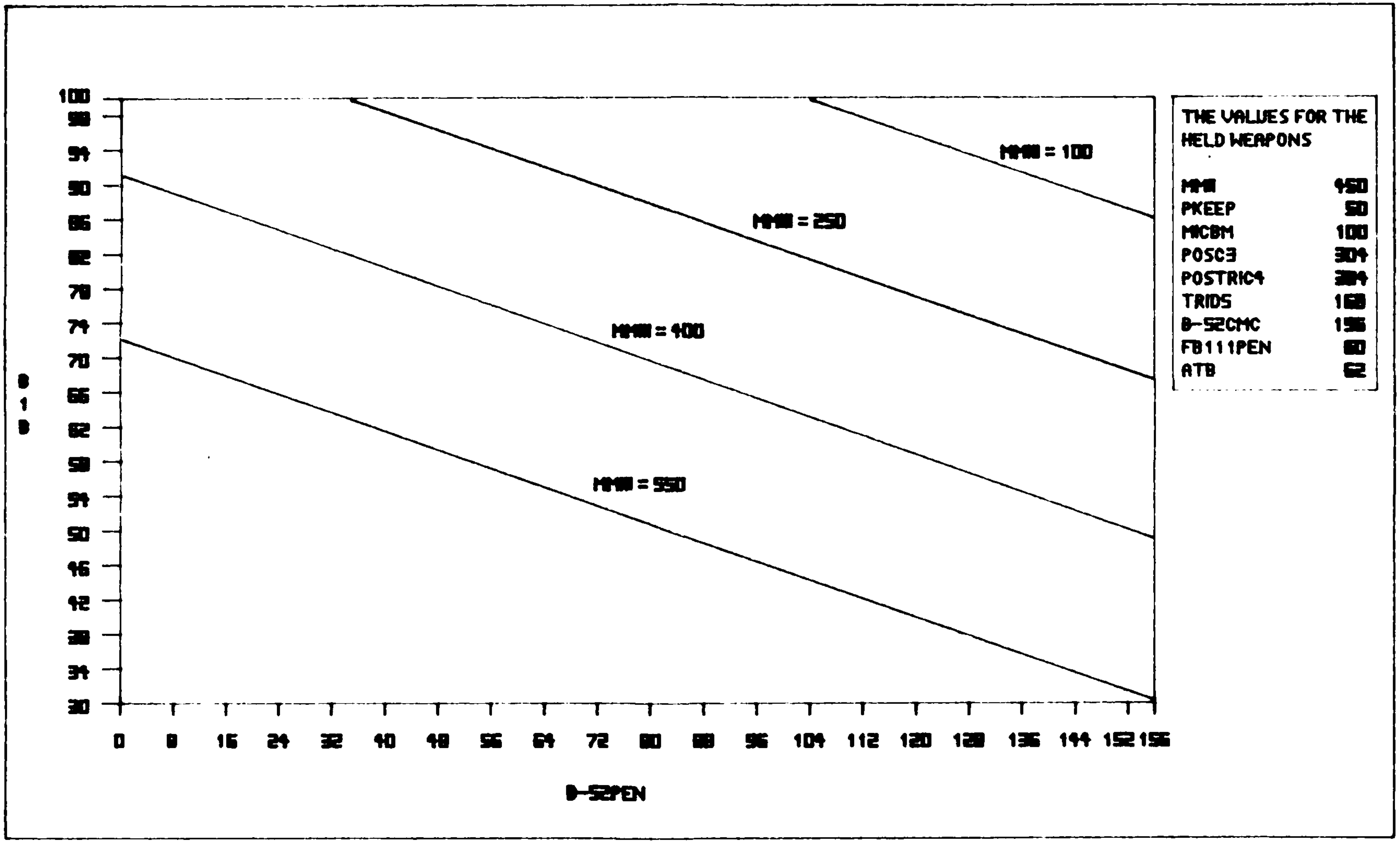


FIG 5.5B: CONSTANT 87% DE CONTOUR LINES FOR VARIOUS COMBINATIONS OF B1BS, B-52PENS, AND MMIIS. ALL OTHER WEAPONS HELD CONSTANT.

estimated model. If not, apparently misleading information may be extracted from the reduced estimated model.

Conclusion

This chapter explores the advantages of RSM: predictability and informational analysis. Predictability was actually shown in Chapter Four under the Validation of the Estimated Model Step. Therefore, this chapter concentrates on explaining how informational analysis is performed.

Usually, the full estimated model is reduced to "workable" terms, if possible. The RGRSSION allows the user to reduce the full estimated model. Since orthogonal designs are used to gather information for estimating the postulated model, the estimated model's coefficients are easily used to illuminate relationships between themselves and with the response. Many graphs are developed to show some of the relationships and how easily information is garnered from them.

Chapter Six, the next and last chapter, concludes this report by summing up the purpose, the product, and the results of this thesis. Also, Chapter Six lists some observations noted during the work of this thesis and reports some recommendations for further study.

Chapter 6: Conclusions, Observations, and Recommendations

Introduction

The Air Force Wide Mission Area Analysis Division (AF/XOXR-MAA) uses a deterministic model called the Arsenal Exchange Model (AEM) to evaluate US strategic offensive capabilities and to answer "what-if" questions(23:2). The evaluations and analyses are usually done through AEM response comparisons of damage expectancy. The damage expectancy response is calculated by the AEM through the use of many input variables: weapons and/or their characteristics, targets and/or their characteristics, and targeting strategies and priorities. As the AEM calculates damage expectancy, the goal of the AEM is to maximize the response.

There is a need to rapidly answer predictable "what-if" questions about US strategic capabilities using the AEM. By conventional methods, "what-if" analysis or informational analysis with the AEM require the varying of one AEM parameter at a time and then executing the AEM for each parameter change. For AF/XOXR-MAA, The AEM requires approximately three hours for one execution (16). Consequently, informational analysis through conventional methods is very slow.

Response Surface Methodology (RSM) uses mathematical and statistical techniques for analyzing problems that use many variables to compute a response and whose goal is to maximize that response (20:445). A variation of RSM, Smith's RSM variant, is RSM applied specifically to a deterministic model for the purpose of facilitating the use of informational analysis (26;27). Given an explicit scenario, Smith's RSM

variant uses experimental designs, with the characteristics of orthogonality and rotatability, and multiple regression to capture the responses of the deterministic model in a carefully selected postulated model. Using a carefully selected postulated model, built from orthogonal and rotatable experimental designs, extensive informational analysis can be easily performed.

The purpose of this thesis was to automate the application of Smith's RSM variant to the AEM. The purpose, of course, is to improve AF/XOXR-MAA's ability to perform informational analysis in real time. Also, the ability to accurately predict the AEM and to provide more informational analysis are two other reasons.

Conclusions

The advantages of applying Smith's RSM variant to a deterministic model like the AEM are accurate predictions and abundant informational analysis in real time. However, there are other advantages and some disadvantages along with some limitations.

Predictability. The ability of the estimated model (postulated model after its parameter coefficients are estimated) to accurately predict the AEM was shown in Chapter Four under the Estimated Model Validation Step. But, accuracy is determined by the user. The user should know within what bounds (standard deviation) the response of the estimated model should fall for the estimated model to be proclaimed as accurately predicting the AEM response.

From the twenty randomly chosen test points shown in Figure 4.10, the standard deviation of the estimated model's response was +/- 2.479 with a mean deviation of +0.164. Three test points inflated the bounds.

Without those three test points, the accuracy of the estimated model becomes much better because the standard deviation of the estimated model's response would be +/- 1.318 with a mean deviation of -0.551.

However, a user cannot arbitrarily delete a test point to improve the estimated model's ability to predict the AEM's response. The purpose of showing the improved statistics after deleting the three high deviations, is to demonstrate that the ability of the estimated model to capture the AEM output surface (responses) fluctuates over the output surface. The ability will be better in some areas of the surface and worse in others. Therefore, a user should be aware of what point on the output surface he/she is estimating a response for and judge that estimate with prior AEM output near that point and/or with the residuals calculated from nearby design points.

If a user can accept the standard deviation of the estimated model, then the user gains the ability of AEM predictability in real time. Instead of requiring three hours to execute (at AF/XOXR-MAA), the estimated model can predict the AEM's response within two seconds. This advantage can be extremely valuable when an answer is needed immediately.

Informational Analysis. The ability of the estimated model to allow informational analysis to be extracted quickly and more abundantly was demonstrated in Chapter Five. Figures 5.2 through 5.5 showed how the parameter estimates of the estimated model could be used to enlighten the relationships between the factors (independent variables) and the response (dependent variable), to clarify the relative worth of each factor towards the response, and to develop constant damage expectancy (DE) contours given various factor combinations. However, the

user was warned to evaluate any informational analysis with regards to the AEM scenario, such as targeting strategies and priorities, to obtain a more complete picture of the information gained through the analysis.

<u>Further Advantages Plus Disadvantages/Limitations</u>. Taken from Smith and Mellichamp, there are some more advantages to using Smith's RSM variant along with some disadvantages and limitations (27:62,63). Besides the advantage of predictability and abundant informational analysis in real time, Smith's RSM variant also allows insight into the deterministic model itself and can be used with any deterministic model, whether linear or nonlinear, including optimization models.

There are three disadvantages/limitations to using Smith's RSM variant. First, Smith's RSM variant is highly scenario dependent. If a factor or the range of a factor changes, the current estimated model becomes invalid. A new estimated model would have to be constructed to accommodate the changed factor or range. Second, no extrapolation can be performed outside the ranges of the estimated model's variables. Extrapolation outside the factor ranges can lead to erroneous information. Last, large numbers of the deterministic model executions are required to construct postulated models containing many variables. For example, for this thesis it took 193 AEM executions to obtain the necessary data for the construction of the twelve variable postulated model.

Observations

Two observations are noted. One occurred while building the RSMed-AEM program, and the other after the program was tested with the application (Chapter Four). First, part of building any subroutine or section of computer code that purports to accomplish some feat is the

responsibility of assuring the code (verifying) does accomplish the feat. All the main-menu subroutines of the RSMed-ABM program were verified. The verification of the RGRSSION subroutine is reported in Chapter Four. It is from verifying the the RGRSSION subroutine that the first observation occurred.

Verification of the RGRSSION subroutine meant insuring the RGRSSION output was correct. The output from the regression package of SAS was the absolute (Appendices Q and R). Three sets of variables -- three, seven, and twelve -- were used in the verification (only the twelve variable set is reported in this thesis). While comparing the limited regression output of the RGRSSION subroutine to the corresponding output of SAS's regression package for each set of variables, other SAS regression statistics, particularly those used with stochastic models, were behaving as if they could be used with a deterministic model (e.g., mean square error (MSE), adjusted R^2, F-test for the betas (βs), and t-tests for the parameter coefficients). This observation is important because most regression statistics are calculated with variance presupposed, and since deterministic models do not have variance, those statistics are assumed not applicable.

The second observation was regretfully noticed after the test application was nearly complete. In Chapter Three, it was assumed that a B-52 could not carry both penetrating bombs (gravity and SRAMs) and ALCMs at the same time. This meant using two design variables to represent the weapons that a B-52 can carry: B-52PEN and B-52CMC. The maximum number of B-52s was set to be 156 carriers. To insure that the max of 156 B-52s was never exceeded, the experimental design should never have a +1 for both design variables at the same time. Unfortu-

nately, the twelve variable Box-Behnken design does not accomplish this. At the conclusion to the test application, it was noticed that, at times, when the experimental design had both B-52 design variables coded at +1, the AEM scenario was using a maximum of 312 B-52s instead of 156.

Recommendations

Five recommendations for future research are noted. First, the RSMed-AEM program could probably be written more efficiently than it currently is. Second, other types of postulated models could possibly be used with the AEM. Third, a graphics subroutine could be written for the automated RSMed-AEM program that would automatically produce the graphs of Chapter Five. Fourth, the automated RSMed-AEM program could be spliced with the AEM to eliminate the function of Tool Two. Fifth, and last, the applicability of all regression statistics to deterministic models could be decided (refer to the Observations section of this chapter).

Appendix A: Box-Behnken Designs

There are ten Box-Behnken experimental designs ranging from three to sixteen variables (designs for eight, thirteen, fourteen, and fifteen variables are not available). The designs are taken from Box and Behnken's article (1). Each Box-Behnken design uses a certain code scheme. Each code scheme shows a plus or a minus sign. The plus sign corresponds to the high range and the negative sign relates to the low range.

A code scheme is shown first and then those Box-Behnken experimental design(s) that use that code scheme next. Each line of an experimental design, except the last line, shows which variables are affected by the code scheme. Per experimental design line, the values of the variables are changed for each line of change in the code scheme. The last line of each experimental design refers to replacing coding all variables with their medium values. The designs are listed in ascending order:

Code Scheme A:

-	-
+	-
-	+
+	+

For Three Variables:

1	2
1	3
2	3
0	0

For Four Variables:

1	2
3	4
1	4
2	3
1	3
2	4
0	0

For Five Variables:

1	2
3	4
2	5
1	3
4	5
2	3
1	4
3	5
1	5
2	4
0	0

Code Scheme B:

-	-	-
+	-	-
-	+	-
+	+	-
-	-	+
+	-	+
-	+	+
+	+	+

For Six Variables:

1	2	4
2	3	5
3	4	6
1	4	5
2	5	6
1	3	6
0	0	0

For Seven Variables:

4	5	6
1	6	7
2	5	7
1	2	4
3	4	7

1	3	5
2	3	6
0	0	0

For Nine Variables:

1	4	7
2	5	8
3	6	9
1	2	3
4	5	6
7	8	9
1	5	9
3	4	8
2	6	7
1	6	8
2	4	9
3	5	7
1	4	7
2	5	8
3	6	9
0	0	0

Code Scheme C:

+	+	+	+
+	+	+	-
+	+	-	-
+	+	-	+
+	-	-	+
+	-	+	+
-	-	+	+
-	+	+	+
-	+	+	-
-	+	-	-
-	+	-	+
-	-	-	+
-	-	-	-
-	-	+	-
+	-	+	-
+	-	-	-

For Ten Variables:

2	6	7	10
1	2	5	10
2	3	7	8
2	4	6	9
1	8	9	10
3	4	5	10
1	4	7	8
3	5	7	9

1	3	6	9
4	5	6	8
0	0	0	0

For Twelve Variables:

1	2	5	7
2	3	6	8
3	4	7	9
4	5	8	10
5	6	9	11
6	7	10	12
1	7	8	11
2	8	9	12
1	3	9	10
2	4	10	11
3	5	11	12
1	4	6	12
0	0	0	0

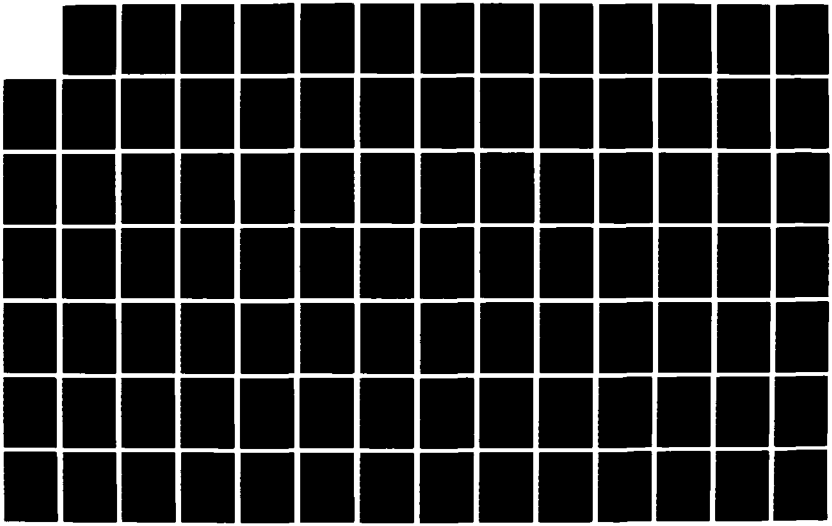
For Sixteen Variables:

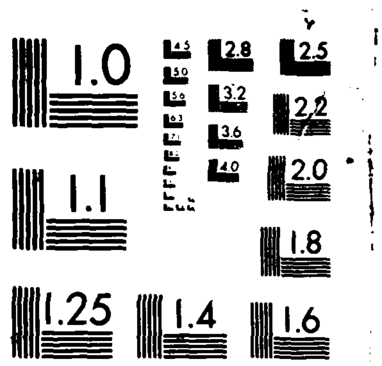
1	2	6	9
3	4	8	11
5	10	13	14
7	12	15	16
2	3	7	10
1	4	5	12
6	11	14	15
8	9	13	16
2	5	6	13
4	7	8	15
1	9	10	14
3	11	12	16
3	6	7	14
1	5	8	16
2	10	11	15
4	9	12	13
1	3	13	15
2	4	14	16
5	7	9	11
6	8	10	12
4	6	10	16
3	5	9	15
1	7	11	13
2	8	12	14
0	0	0	0

Code Scheme D:

- - - - +
+ - - - -

AD-A185 247 AUTOMATING RESPONSE SURFACE METHODOLOGY FOR THE ARSENAL 2/4
EXCHANGE MODEL(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. G E YIELDING
UNCLASSIFIED MAY 87 AFIT/GOR/ENS/86D-17 F/G 12/4 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

-	+	-	-	-
+	+	-	-	+
-	-	+	-	-
+	-	+	-	+
-	+	+	-	+
+	+	+	-	-
-	-	-	+	-
+	-	-	+	+
-	+	-	+	+
+	+	-	+	-
-	-	+	+	+
+	-	+	+	-
-	+	+	+	-
+	+	+	+	+

For Eleven Variables:

3	7	8	9	11
1	4	8	9	10
2	5	9	10	11
1	3	6	10	11
1	2	4	7	11
1	2	3	5	8
2	3	4	6	9
3	4	5	7	10
4	5	6	8	11
1	5	6	7	9
2	6	7	8	10
0	0	0	0	0

Appendix B: The RSMed-AEM Program

```
c Title
c
c
c Automation of RSM to AEM
c
c
      Program RSMedAEM
c
c
      call menu
      stop
      end
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                         c
c                        Functions and Subroutines                        c
c                                                                         c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                         c
      Subroutine Menu
c                                                                         c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c     Purpose:  Subroutine Menu controls branching within this program.  It
c               allows the user to select what he/she wants to do.  The options
c               available are:
c
c               (I)  Input Data -- input the number of variables and their
c                                  quantities.
c
c               (U)  Build Uncoded Design -- build the uncoded design that is
c                                            used as input to the AEM.
c
c               (C)  Build Coded Design -- build coded design that is used with
c                                          a regression package to calculate the
c                                          regression equation's coefficients.
c
c               (R)  Regression Analysis -- perform regression analysis on
c                                           the AEM output.
c
```

```
c               (D)  Decode -- decode the estimated regression equation's
c                              coefficients calculated under (C) above.
c
c               (E)  Exit -- exit this program.
c
c
c     Variables:
c
c
      character*1 finished,readcr,dirction,code
      data finished/'n'/, readcr/' '/, dirction/' '/
c
c
c     Begin:
c
c
 10   if (finished .eq. 'n') then
        call scrlscrn(24)
        print 100
100     format (' ********************************** MAIN MENU *****',
     &          '******************************')
        call scrlscrn(4)
        print *, '                          (I)  Input Data'
        call scrlscrn(1)
        print *, '                          (U)  Build Uncoded Design'
        call scrlscrn(1)
        print *, '                          (C)  Build Coded Design'
        call scrlscrn(1)
        print *, '                          (R)  Regression'
        call scrlscrn(1)
        print *, '                          (D)  Decode'
        call scrlscrn(1)
        print *, '                          (X)  Execute Equations'
        call scrlscrn(1)
        print *, '                          (E)  Exit Program'
        call scrlscrn(3)
        print *, ' Please enter a letter: '
        call scrlscrn(1)
        read (*,200) dirction
200     format (a)
        if (dirction .eq. 'I' .or. dirction .eq. 'i') then
          call InptData
        else
          if (dirction .eq. 'U' .or. dirction .eq. 'u') then
            code = 'n'
            call CodeDsgn(code)
          else
            if (dirction .eq. 'C' .or. dirction .eq. 'c') then
              code = 'y'
              call CodeDsgn(code)
            else
              if (dirction .eq. 'R' .or. dirction .eq. 'r') then
                call Rgrssion
```

```
              else
                if (dirction .eq. 'D' .or. dirction .eq. 'd') then
                  call Decode
                else
                  if (dirction .eq. 'X' .or. dirction .eq. 'x') then
                    call Xcuteqns
                  else
                    if (dirction .eq. 'E' .or. dirction .eq. 'e') then
                      finished = 'y'
                    else
                      call scrlscrn(11)
                      print 300
                      call scrlscrn(11)
                      read (*,200) readcr
                    endif
                  endif
                endif
              endif
            endif
          endif
        endif
        go to 10
      endif
 300  format (' ','Bad input!!  Press "return" to try again:')
      return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine InptData
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  Subroutine InptData is the "controller" subroutine for
c                 entering all data pertaining to the AEM.
c
c
c       Variables:
c
c
c       paramtrs -- the number of design paramters.
c       external -- flag that reflects whether or not the parameter's data
c                   already resides on an EXTERNAL data set.
c       goodchar -- flag that reflects whether or not user input for yes-no
c                   questions is an acceptable character.
c       exit     -- flag that allows the user to exit this "main" subroutine
c                   if there is an unrecoverable error or, indirectly through
c                   the "getdata" variable, if an experimental design is
c                   unavailable.
c       getdata  -- flag that allows the user to input the parameter's
```

```
c                    characteristics if an experimental design can be made of
c                    use.
c
c
      integer*2 paramtrs
      common /parms/paramtrs
      character*1 external,goodchar,exit,getdata,code
c
c
c       Begin:
c
c
c       Ask the user if he wishes to use an outside source to enter the
c       number of design parameters and their characteristics.  Make sure
c       the user responds correctly.
c
c
    1 call ScrlScrn(11)
      print *, 'Does data already reside on external data set?  (y/n)'
      call ScrlScrn(11)
      read (*,2) external
    2 format (a)
      call ChckChar(external,goodchar)
      if (goodchar .ne. 'y') then
        call BadChar
        go to 1
      endif
c
c
c       If user wishes to use external data, branch to Subroutine XtrnlDat.
c       If not, then the user wishes to input the data via the scope.
c       Therefore, start the interactive input by branching to
c       Subroutine GetParms.
c
      if (external .eq. 'y' .or. external .eq. 'Y') then
        call XtrnlDat(exit)
      else
        call GetParms(getdata)
c
c
c       If the user has found an adequate design for the number of parameters,
c       branch to Subroutine AskForIt to interactively input each parameter's
c       characteristic.  If not, set the "exit" variable appropriately to
c       exit this subroutine and back to the Main Menu.
c
c
        if (getdata .eq. 'y') then
          exit = 'n'
          call AskForIt
        else
          exit = 'y'
        endif
      endif
```

```
c
c
c       If there is not an unrecoverable error or the user does not wish to
c       exit the system, branch to Subroutine VerfyDat to allow the user to
c       verify, change, or correct each parameter's characteristics.  Next,
c       branch to Subroutine SaveIt to allow the user the chance to save
c       the parameter data.
c
c
      if (exit .eq. 'n') then
        call VerfyDat
        call SaveIt
      endif
c
c
c       Mission complete:
c
c
      return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine XtrnlDat(baddata)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  Subroutine XtrnlDat uses an external data set to enter the
c                 design variables' characteristics.
c
c
c       Variables:
c
c
ccc     COMMON VARIABLES:
c
c
c       paramtrs -- the number of design paramters.
c       varnames -- the name associated with each parameter.
c       controls -- the number of AEM variables controlled by each design
c                   parameter.
c       mid      -- mid point between the low and high value for each
c                   parameter.
c       div      -- the difference between the low and mid points for each
c                   parameter which is the same as the difference between
c                   the mid and high points for each parameter.
c        low     -- the low value for each parameter.
c       high     -- the high value for each parameter.
c
c
```

```
ccc
c
c
c       baddata  -- a flag to notify the Subroutine InptData that the input
c                   data from the external data set is bad.
c       readcr   -- this variable does nothing more than accept user "returns"
c                   or other superficial user input.
c       i        -- a counter for DO loops.
c
c
      integer*2 paramtrs
      common /parms/paramtrs
      integer*2 controls(16)
      double precision mid(16),div(16),low(16),high(16)
      common /uncode/mid,div,low,high,controls
      character*8 varnames(16)
      common /names/varnames
      character*1 baddata,readcr
      integer*2 i
c
c
c       Begin:
c
c
c       Initialize the following variables and open the external file for
c       reading.  If there is an error openning the file, notify the user.
c
c
      i = 0
      baddata = 'n'
      paramtrs = 0
      open (unit=4,file='varibles.dat',status='old',err=8)
c
c
c       Read in the data from the external data set.  If an error occurs,
c       inform the user and exit this subroutine.
c
c
    1 i = i + 1
      read (4,2,end=3,err=5) varnames(i),low(i),high(i),controls(i)
    2 format (2X,A8,2X,F7.2,2X,F7.2,2X,I2)
      div(i) = (high(i) - low(i)) / 2.0
      mid(i) = (low(i) + high(i)) / 2.0
      paramtrs = paramtrs + 1
      go to 1
c
c
c       Check the data for logic error.  Logic errors will occur if the
c       the data is out of range.  If there is an error, set the appropriate
c       flag which notifies the user of it.
c
c
    3 do 4 i = 1,paramtrs
```

```
c
c
c       Is the number of design parameters within range?
c
c
        if (paramtrs .lt. 3 .or. paramtrs .gt. 16) then
          baddata = 'y'
        endif
        if (paramtrs .eq. 8 .or. paramtrs .eq. 13 .or.
     &      paramtrs .eq. 14 .or. paramtrs .eq. 15) then
          baddata = 'y'
        endif
c
c
c       Are the low values for each design variable within range?
c
c
        if (low(i) .lt. 0 .or. low(i) .gt. 9999) then
          baddata = 'y'
        endif
c
c
c       Are the high values for each design variable within range?
c
c
        if (high(i) .lt. low(i) .or. high(i) .gt. 9999) then
          baddata = 'y'
        endif
c
c
c       Are the number of AEM variables controlled by each design within range?
c
c
        if (controls(i) .lt. 1 .or. controls(i) .gt. 4) then
          baddata = 'y'
        endif
    4  continue
c
c
c       If data is bad, inform the user and insure the "baddata" flag is set
c       before exiting this subroutine.
c
c
       if (baddata .eq. 'y') then
    5   baddata = 'y'
        call ScrlScrn(11)
        write (*,6)
    6   format (' There is a problem with the external input.  Please ',
     &  'exit this program and'/' insure that the input data is ',
     &  'correct.  Press "return" to continue.')
        call ScrlScrn(10)
        read (*,7) readcr
    7   format (a)
```

```
      endif
c
c
c       Close the external data set and branch to the "subroutine exit code".
c
c
      close(4)
      go to 10
c
c
c       Error code for openning the external data set.
c
c
    8 call ScrlScrn(11)
      write (*,9)
    9 format (' An error has occured from trying to open data set ',
     &'"Varibles.Dat".  Please '/' correct the problem or reenter ',
     &'Subroutine InptData and enter the data via the'/' scope.  ',
     &'Press "return" to continue.')
      call ScrlScrn(9)
      read (*,7) readcr
      baddata = 'y'
c
c
c       Mission complete:
c
c
   10 return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine GetParms(getdata)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  If the user wishes to enter design variable information
c                 via the scope, find out first how many design variables
c                 are of interest.
c
c
c       Variables:
c
c
c       paramtrs -- the number of design paramters.
c       doagain  -- a flag that relates the user's wish to try another
c                   experimental design if there is no design, currently,
c                   for the number of design variables he/she is
c                   interested in.
c       getdata  -- a flag that is sent back to Subroutine InptData that
```

```
c                    relates whether or not the user found an appropriate
c                    design number parameter.
c       correct  -- a fail-safe flag to insure the user is satisfied with
c                    his/her choice for the design variable parameter.
c       goodchar -- flag that reflects whether or not user input for yes-no
c                    questions is an acceptable character.
c
c
      integer*2 paramtrs
      common /parms/paramtrs
      character*1 doagain,getdata,correct,goodchar
c
c
c       Begin:
c
c
c       Initialize the following variables:
c
c
      getdata = 'y'
      correct = 'n'
c
c
c       This whole subroutine is nothing more than a WHILE loop.  The loop will
c       continue until the user determines whether or not he/she can use what
c       design variable parameters are available.
c
c
    1 if (correct .eq. 'n' .or. correct .eq. 'N') then
c
c
c       Allow the user to input the number of design variables of interest.
c       Check for input error and then initialize the "doagain" variable.
c
c
        call ScrlScrn(11)
        print *, ' How many variables are there?  (Max of 16)'
        call ScrlScrn(11)
        read (*,*,err=7) paramtrs
        call ScrlScrn(24)
        doagain = 'y'
c
c
c       If the number entered by the user is not within bounds, inform the
c       user and check whether or not the user wishes to try another design.
c       Check the user's input for the correct possible characters and allow
c       the user to re-answer if he/she made an input mistake.  If the user
c       wishes to retry, allow him/her to reenter the value for the design
c       variable parameter.  If not, set the appropriate flag to exit this
c        subroutine.
c
c
    2   if (paramtrs .lt. 3 .or. paramtrs .gt. 16) then
```

```
          print 3
    3     format (' Value has to be between 3 and 16, inclusively!!  ',
     &    'Do you wish to retry?  (y/n)')
          call ScrlScrn(11)
          read (*,4) doagain
    4     format (a)
          call ChckChar(doagain,goodchar)
          if (goodchar .ne. 'y') then
            call BadChar
            go to 2
          endif
        else
c
c
c       If the user's value for the design variable parameter is within range,
c       make sure his/her value is for one of the experimental designs that is
c       currently available.  If it isn't, again allow the user to retry if
c       he/she wishes while checking his/her input for the correct possible
c       character.
c
c
          if (paramtrs .eq. 8 .or. paramtrs .eq. 13 .or.
     &        paramtrs .eq. 14 .or. paramtrs .eq. 15) then
            print 5, paramtrs
    5       format (' Sorry, this program currently has no experimental'
     &              ' design for ',I2,' paramters.'/' Do you wish to'
     &              ' reenter another value for parameters?  (y/n)')
            call ScrlScrn(10)
            read (*,4) doagain
            call ChckChar(doagain,goodchar)
            if (goodchar .ne. 'y') then
              call BadChar
              go to 2
            endif
          else
c
c
c       If the user got by the first two checks, make sure that he/she is
c       satisfied with the selection.  Check his/her answer for error.  If
c       the user likes what he/she has, set the appropriate flag to exit this
c       subroutine.  If not, allow the user to reenter by again setting the
c       appropriate flag.
c
c
            print 6, paramtrs
    6       format (' Is parameters = ',I2,' correct?  (y/n)')
            call ScrlScrn(11)
            read (*,4) correct
            call ChckChar(correct,goodchar)
            if (goodchar .ne. 'y') then
              call BadChar
              go to 2
            endif
```

```
          end if
        endif
c
c
c       Allow the user to retry if he/she wishes to do so after failing the
c       bounds and designs available checks.  If the user did not wisht to
c       retry, set the appropriate flag and exit this subroutine.
c
c
        if (doagain .eq. 'y' .or. doagain .eq. 'Y') then
          go to 1
        else
          getdata = 'n'
        endif
c
c
c       End WHILE loop and branch to subroutine exit code.
c
c
      end if
      go to 8
c
c
c       Error code branching.
c
c
    7 call BadInput
      go to 1
c
c
c       Mission accomplished:
c
c
    8 return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine AskForIt
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  Subroutine AskForIt interactively queries the user for
c                 each design variable's characteristics (i.e., name, low
c                 and high values, and the number of AEM variables controlled.
c
c
c       Variables:
c
c
```

```
ccc     COMMON VARIABLES -- refer to Subroutine XtrnlDat for the definitions
c                   of the "common" variables.
c       readcr   -- this variable does nothing more than accept user "returns"
c                   or other superficial user input.
c       i        -- a counter for DO loops.
c
c
      integer*2 paramtrs
      common /parms/paramtrs
      character*8 varnames(16)
      common /names/varnames
      integer*2 controls(16)
      double precision mid(16),div(16),low(16),high(16)
      common /uncode/mid,div,low,high,controls
      character*1 readcr
      integer*2 i
c
c
c       Begin:
c
c
c       Before obtaining each design variable's characteristics give the
c       user some instructions on how to input the data.
c
c
      call ScrlScrn(5)
      print 1
    1 format ('      The following set of questions will ask you for',
     &' the variable''s name, its ',/,
     &' low and high values, and the number of AEM input variables ',
     &'it controls.  The ',/,
     &' limit to each variable''s characteristic is:',///,
     &'      Name                 -- up to eight characters',//,
     &'      Low Value            -- up to a value of 9999',//,
     &'      High Value           -- up to a value of 9999',//,
     &'      Number of AEM Input',/,
     &'      Variables Controlled -- up to a value of 4',///,
     &' When you are ready to continue, press "return" :')
      call ScrlScrn(3)
      read (*,2) readcr
    2 format (a)
c
c
c       Obtain each design variable's name, low and high value, and the
c       number of AEM input variables it controls.
c
c
      do 14 i = 1,paramtrs
c
c
c       What is the name associated with this design variable?
c
c
```

```
      call ScrlScrn(11)
      print 3, i
    3 format (' What is the name for variable ',i2,'?')
      call ScrlScrn(11)
      read (*,4), varnames(i)
    4 format (a8)
c
c
c     What is the low value for this design variable?  Check for input
c     error and insure the low value is not less than zero.  If either
c     happens, let the user reenter the value.
c
c
    5 call ScrlScrn(11)
      print 6, varnames(i)
    6 format (' What is the low value for variable "',a8,'"?')
      call ScrlScrn(11)
      read (*,*,err=15) low(i)
      if (low(i) .lt. 0) then
        call ScrlScrn(11)
        print 7
    7   format (' Low value cannot be less than zero!!  Press ',
     &  '"return" and try again.')
        call ScrlScrn(11)
        read (*,2) readcr
        go to 5
      endif
c
c
c     What is the high value for this design variable?  Check for input
c     error and insure the high value is not less than the low value.
c     If either happens, let the user reenter the value.
c
c
    8 call ScrlScrn(11)
      print 9, varnames(i)
    9 format (' What is the high value for variable "',a8,'"?')
      call ScrlScrn(11)
      read (*,*,err=16) high(i)
      if (high(i) .lt. low(i)) then
        call ScrlScrn(11)
        print 10
   10   format (' High value cannot be less than low value!!  ',
     &  'Press "return" and try again.')
        call ScrlScrn(11)
        read (*,2) readcr
        go to 8
      endif
c
c
c     How many ABM variables are controlled by this design variable.
c     Again, check for input error and insure the that the value entered
c     is within the bounds.  If input is bad, allow the user to reenter
```

```
c       the value.
c
c
 11     call ScrlScrn(11)
        print 12, varnames(i)
 12     format (' How many AEM input variables does variable "',a8,
     &  '" control?')
        call ScrlScrn(11)
        read (*,*,err=17) controls(i)
        if (controls(i) .lt. 1 .or. controls(i) .gt. 4) then
          call ScrlScrn(11)
          print 13
 13       format (' Value has to be between 1 and 4 inclusively!!  ',
     &    'Press "return" and try again.')
          call ScrlScrn(11)
          read (*,2) readcr
          go to 11
        endif
c
c
c       Compute the "mid" and "div" values for each design variable.  The
c       "mid" and "div" points are used for obtaining the coded variables
c       in the experimental design section of this program.
c
c
        div(i) = (high(i) - low(i)) / 2.0
        mid(i) = low(i) + div(i)
 14   continue
c
c
c       Branch to subroutine exit code.
c
c
      go to 18
c
c
c       Error checking code.
c
c
 15   call BadInput
      go to 5
 16   call BadInput
      go to 8
 17   call BadInput
      go to 11
c
c
c       Beam me up, Scottie.
c
c
 18   return
      end
c
```

```
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine VerfyDat
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c        Purpose:  Subroutine VerfyDat allows the user to verify, modify, or
c                  change each parameter's characteristics.
c
c
c        Variables:
c
c
ccc      COMMON VARIABLES -- refer to Subroutine XtrnlDat for the definitions
c                     of the "common" variables.
c        chngdata -- flag that controls when the user wants to modify parameter
c                     characteristics.
c        correct  -- flag that controls when the user is through with this
c                     subroutine.
c        readcr   -- this variable does nothing more than accept user "returns"
c                     or other superficial user input.
c        goodchar -- flag that reflects whether or not user input for yes-no
c                     questions is an acceptable character.
c
c
      integer*2 paramtrs
      common /parms/paramtrs
      character*8 varnames(16)
      common /names/varnames
      integer*2 controls(16)
      double precision mid(16),div(16),low(16),high(16)
      common /uncode/mid,div,low,high,controls
      double precision newvalue
      integer*2 i,j
      character*1 chngdata,correct,readcr,goodchar
      character*8 newvarnm
c
c
c        Begin:
c
c
c        Initialize the following variables:
c
c
      chngdata = 'n'
      correct = 'n'
c
c
c        This whole subroutine is nothing more than a WHILE loop.  If data is
c        not correct, as it is assumed not to be when the loop is first entered,
```

```
c       the user is shown his/her data and queried if it is correct or not.  If
c       user accepts it, the loop and thus this subroutine is exited, else the
c       user is allowed to modify his/her data.
c
c
   1  if (correct .eq. 'n' .or. correct .eq. 'N') then
   2    call InpTable
c
c
c       If the user wants to change the input, ask for which row and column
c       the bad data is located on and then for the new value.  Check the
c       user's input for validity and allow him/her to reenter if necessary.
c
c
        if (chngdata .eq. 'y') then
c
c
c       Which row?  Make sure user input is within range.
c
c
          print *, ' Which row is the bad data located on?  '
          read (*,*,err=15) i
          if (i .lt. 1 .or. i .gt. paramtrs) then
            call ScrlScrn(11)
            print 3, paramtrs
   3        format (' Bad input!!  Rows range from 1 to ',i2,
     &      '.  Press "return" to retry.')
            call ScrlScrn(11)
            read (*,4) readcr
   4        format (a)
            go to 2
          endif
c
c
c       Which column?  Make sure user inputs within the correct range.
c
c
   5      call InpTable
          print *, ' Which column is the bad data located on?  '
          read (*,*,err=16) j
          if (j .lt. 1 .or. j .gt. 4) then
            call ScrlScrn(11)
            print 6
   6        format (' Bad input!!  Columns range from 1 to 4.  Press'
     &      ' return to retry.')
            call ScrlScrn(11)
            read (*,4) readcr
            go to 5
          endif
c
c
c       Receive the new value from the user and check it for error.  Allow
c       the user to reenter the data if not entered correctly.
```

```
c
c
 7      call InpTable
        print *, ' What is the new value?  '
c
c
c      Change the "name" of the variable.
c
c
        if (j .eq. 1) then
          read (*,8), newvarnm
 8        format (a8)
          varnames(i) = newvarnm
        else
c
c
c      Change either the low, high, or the number of ABM variables
c      controlled by the design variable.  Check user input for error.
c      If error occurs, allow the user to reenter.
c
c
          read (*,*,err=17), newvalue
c
c
c      Change the low value.
c
c
          if (j .eq. 2) then
c
c
c      Do not allow the low value to be less than zero.  If it is reenter,
c      else accomplish the change.
c
c
            if (newvalue .lt. 0) then
              call ScrlScrn(11)
              print 9
 9            format (' Low value cannot be less than zero!!  Press ',
  &           '"return" and try again.')
              call ScrlScrn(11)
              read (*,4) readcr
              go to 7
            else
              low(i) = newvalue
              div(i) = (high(i) - low(i)) / 2.0
              mid(i) = low(i) + div(i)
            endif
          endif
c
c
c      Change the high value.
c
c
```

```
          if (j .eq. 3) then
c
c
c      Do not allow the high value to be less than the variables low value.
c      If it is reenter; else accomplish the change.
c
c
            if (newvalue .lt. low(i)) then
              call ScrlScrn(11)
              print 10
  10          format (' High value cannot be less than low value!!  ',
     &        'Press "return" and try again.')
              call ScrlScrn(11)
              read (*,4) readcr
              go to 7
            else
              high(i) = newvalue
              div(i) = (high(i) - low(i)) / 2.0
              mid(i) = low(i) + div(i)
            endif
          endif
c
c
c      Change the number of AEM variables controlled by the design variable.
c
c
          if (j .eq. 4) then
c
c
c      Value has to be between the numbers 1 and 4 inclusively.  If not,
c      allow the user to reenter; else, accomplish the change.
c
c
            if (newvalue .lt. 1 .or. newvalue .gt. 4) then
              call ScrlScrn(11)
              print 12
  12          format (' Value has to be between 1 and 4 inclusively!'
     &        '  Press "return" and try again.')
              call ScrlScrn(11)
              read (*,4) readcr
              go to 7
            else
              controls(i) = int(newvalue)
            endif
          endif
        endif
c
c
c      Change the "chngdata" flag back to "no" after the user has changed
c      the data.  This allows the recently modified input data to be shown
c      before, if necessary, any further changes are entered.
c
c
```

```
          chngdata = 'n'
        else
c
c
c       Show the user his/her input and ask if it is correct.  This allows
c       the user to modify or change the data if he/she wants to.  Make
c       sure the user inputs the correct answer.
c
c
  13      print 14
  14      format (/' Is the input to all variables correct?  (y/n)')
          read (*,4) correct
          call ChckChar(correct,goodchar)
          if (goodchar .ne. 'y') then
            call BadChar
            call InpTable
            go to 13
          endif
c
c
c       If data is not correct, set the "chngdata" flag to allow the user
c       to change it.
c
c
          if (correct .eq. 'n' .or. correct .eq. 'N') then
            chngdata = 'y'
          endif
        end if
        go to 1
c
c
c       End WHILE loop.
c
c
      endif
c
c
c       Branch to end of this subroutine.
c
c
      go to 18
c
c
c       The following are the error branching calls made from the preceeding
c       "read" statements that "read-in" user input.
c
c
   15  call BadInput
      go to 2
  16  call BadInput
      go to 5
  17  call BadInput
      go to 7
```

```
c
c
c      Mission complete:
c
c
  18  return
      end
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                               c
      Subroutine SaveIt
c                                                                               c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c      Purpose:  Allow the user to save his/her input if so desired.
c
c
c      Variables:
c
c
ccc    COMMON VARIABLES -- refer to Subroutine XtrnlDat for the definitions
c                  of the "common" variables.
c      save     -- a flag that reflects the user's desire to save his/her
c                  data on an externa file.
c      retry    -- a flag that reflects the user's desire to try and save
c                  his/her data after an error has occured from trying to
c                  open the external data set.  It is assumed that the user
c                  can correct the problem outside this program without
c                  having to exit this program, thereby losing all data.
c      readcr   -- this variable does nothing more than accept user "returns"
c                  or other superficial user input.
c      goodchar -- flag that reflects whether or not user input for yes-no
c                  questions is an acceptable character.
c      i        -- a counter for DO loops.
c
c
      integer*2 paramtrs
      common /parms/paramtrs
      character*8 varnames(16)
      common /names/varnames
      integer*2 controls(16)
      double precision mid(16),div(16),low(16),high(16)
      common /uncode/mid,div,low,high,controls
      character*1 save,retry,readcr,goodchar
      integer*2 i
c
c
c      Begin:
c
c
```

```
c       Write the value of the variables 'paramtrs' and 'varnames' to an
c       external data set to be used with the RGRSSION subroutine.
c
c
      open (unit=2,name='rgrssion.in',status='old')
      open (unit=3,name='decode.in',status='old')
      open (unit=4,name='codedsgn.in',status='old')
      write (2,13) paramtrs
      write (4,13) paramtrs
      do 14 i = 1,paramtrs
        write (2,15) varnames(i)
        write (3,16) varnames(i),low(i),high(i),mid(i),div(i)
        write (4,17) low(i),mid(i),high(i),controls(i)
   14 continue
   13 format (' ',I2)
   15 format (' ',A8)
   16 format (' ',A8,2(1X,F7.2),2(1X,F11.6))
   17 format (' ',3(1X,F7.2),1X,I1)
      close (2)
      close (3)
      close (4)
c
c
c       Find out if the user would like to save his/her data.  Make sure
c       the user inputs one of the possible answers.
c
c
    1 call ScrlScrn(11)
      write (*,2)
    2 format (' Do you wish to save your input (y/n)?  If so, the data',
     &' is written to file'/' "varibles.dat".  Consequently, any ',
     &'data already in that file is written over.'/' Input (y/n):')
      call ScrlScrn(9)
      read (*,3) save
    3 format (a)
      call ChckChar(save,goodchar)
      if (goodchar .ne. 'y') then
        call BadChar
        go to 1
      endif
c
c
c       If the user wants to save his/her data, open the file and write
c       the data to it.  Check for errors from openning the file and from
c       writing the data to it.  When finished, close the file.
c
c
      if (save .eq. 'y' .or. save .eq. 'Y') then
        open (unit=4,file='varibles.dat',status='old',err=7)
        do 5 i = 1,paramtrs
          write (4,4,err=10) varnames(i),low(i),high(i),controls(i)
    4     format (2X,A8,2X,F7.2,2X,F7.2,2X,I2)
    5   continue
```

```
    6    close(4)
       endif
c
c
c        Check out of this place.
c
c
       go to 12
c
c
c        Error checking code:
c
c
c        If there is a problem openning the external data set, notify the
c        user and see if he wants to retry.  Check his input for error and
c        allow him/her to reenter his/her answer if there is an error.
c
c
    7  call ScrlScrn(11)
       write (*,8)
    8  format (' There is a problem openning data set "varibles.dat".',
      &/' Can you correct and retry?  (y/n):')
       call ScrlScrn(10)
       read (*,3) retry
       call ChckChar(retry,goodchar)
       if (goodchar .ne. 'y') then
         call BadChar
         go to 7
       endif
c
c
c        If the user wants to retry, allow him/her to do so.  However, it
c        is assumed that before the retry, the user will correct the problem
c        outside of this program.  If the user does not wish to retry, inform
c        him/her of the data loss and exit this subroutine.
c
c
       if (retry .eq. 'y' .or. retry .eq. 'Y') then
         go to 1
       else
         call ScrlScrn(10)
         write (*,9)
    9    format (' Data was not saved unless it previously resided on',
      &  ' the external data set.'/' Press "return" to continue.')
         call ScrlScrn(11)
         read (*,3) readcr
         go to 12
       endif
c
c
c        This error checking code informs the user that an error occured
c        while trying to write the data to the external data set.  The
c        file is closed and then this subroutine is then exited.
```

```
c
c
  10  call ScrlScrn(10)
      write (*,11)
  11  format (' An error has occured from trying to write the data to',
     &' the external data set.'/' Data can not be saved!!!  Examine ',
     &'the external data set for possible clues.'/' Press',
     &' "return" to continue.')
      call ScrlScrn(10)
      read (*,3) readcr
      go to 6
c
c
c       Finished; Capice!!!
c
c
  12  return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                      c
      Subroutine ScrlScrn(scrlines)
c                                                                      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  Subroutine ScrlScrn scrolls the terminal screen by rolling
c                 24 lines across it.
c
c
c       Variables:
c
c
c     i        -- counter for DO loop
c     scrlines -- number of lines to scroll
c
c
      integer*2 i,scrlines
c
c
c       Begin:
c
c
c       Scroll the number of times as dictated by the passed parameter.
c
c
      if (scrlines .gt. 0) then
        do 1 i = 1,scrlines
          print *, ' '
   1    continue
      endif
```

```
      return
      end
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
      Subroutine ChckChar(varible,goodchar)
c                                                                       c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  Subroutine ChckChar evaluates whether or not the passed
c                 parameter "varible"'s value is one of four possible
c                 characters.  The passing subroutine is informed of the
c                 results of the evaluation.
c
c
c       Variables:
c
c
c       varible  -- contains the value passed by the inquisitive subroutine
c       goodchar -- flag that reflects whether or not user input for yes-no
c                   questions is an acceptable character.
c
c
      character*1 varible,goodchar
c
c
c       Begin:
c
c
c       Evaluate the passed parameter and set "results" flag, "goodchar",
c       to the appropriate value.
c
c
      if (varible .eq. 'y' .or. varible .eq. 'Y') then
        goodchar = 'y'
      else
        if (varible .eq. 'n' .or. varible .eq. 'N') then
          goodchar = 'y'
        else
          goodchar = 'n'
        endif
      endif
c
c
c       Leave this tiny subroutine:
c
c
      return
      end
c
```

```
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine BadChar
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  Inform the user that he/she input the wronge character.
c
c
c       Variables:
c
c
      character*1 readcr
c
c
c       Begin:
c
c
      call ScrlScrn(24)
      print 1
   1  format (' Wronge character!!!  Press "return" to try again.')
      call ScrlScrn(11)
      read (*,2) readcr
   2  format (a)
      return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine BadInput
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:
c
c
c       Variables:
c
c
c       readcr   -- this variable does nothing more than accept user "returns"
c                   or other superficial user input.
c
c
      character*1 readcr
c
c
c       Begin:
```

```
c
c
c       Inform the user of his/her blunder and then await his/her
c       understanding nod.
c
c
      call ScrlScrn(11)
      print 1
    1 format (' Bad data:  Either value entered was too large or non-',
     &'numeric!!!  Press "return"',/,' and try again.')
      call ScrlScrn(10)
      read (*,2) readcr
    2 format (a)
c
c
c       I'm out of here:
c
c
      return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine InpTable
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  Subroutine InpTable shows the user each design variable's
c                 characteristic in a formatted style.
c
c
c       Variables:
c
c
ccc     COMMON VARIABLES -- refer to Subroutine XtrnlDat for the definitions
c                   of the "common" variables.
c       i        -- a counter for DO loops.
c
c
      integer*2 paramtrs
      common /parms/paramtrs
      integer*2 controls(16)
      double precision mid(16),div(16),low(16),high(16)
      common /uncode/mid,div,low,high,controls
      character*8 varnames(16)
      common /names/varnames
      integer i
c
c
c       Begin:
```

```
c
c
c       Show the user his/her input.
c
c
      call ScrlScrn(24)
      print 1
    1 format (' Your input for each variable is:'//
     &5X,'Variable',13X,'Range',20X,'Number Of AEM'/
     &20X,'Low',10X,'High',7X,'Input Variables Controlled'/)
      do 3 i = 1,paramtrs
        print 2, varnames(i),low(i),high(i),controls(i)
    2   format (5X,A8,5X,F7.2,7X,F7.2,17X,I2)
    3 continue
c
c
c       Bye, Bye, Love:
c
c
      return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
       Subroutine CodeDsgn(code)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:
c
c
c       Variables:
c
c
      real dsgnmtrx(385,64)
      common /dsgnmtrx/dsgnmtrx
      integer*2 paramtrs,controls(16)
      double precision low(16),mid(16),high(16)
      common /uncode/paramtrs,low,mid,high,controls
      integer*2 features(3:16,3)
      common /features/features
      character*1 code
      integer*2 i,j,totlrows,totlcols
      data features/ 3, 6,10, 6, 7, 0,15,10,11,12, 0, 0, 0,24,
     &               4, 4, 4, 8, 8, 0, 8,16,16,16, 0, 0, 0,16,
     &               2, 2, 2, 3, 3, 0, 3, 4, 4, 4, 0, 0, 0, 4/
c
c       Begin:
c
c
```

```
      open (unit=4,name='codedsgn.in',status='old')
      read (4,375), paramtrs
      do 3 i = 1,paramtrs
        read (4,400) low(i),mid(i),high(i),controls(i)
    3 continue
      close (4)
      if (code .eq. 'y') then
        open(unit=7,file='dsgnmtrx.in',status='old')
      else
        open(unit=8,file='aem.dat',status='old')
      endif
      call SetUp
      call Design3(code)
      totlrows = features(paramtrs,1)*features(paramtrs,2) + 1
      do 1 i = 1,paramtrs
        totlcols = totlcols + controls(i)
    1 continue
      if (code .eq. 'y') then
        write (7,325) features(paramtrs,1),features(paramtrs,2)
      endif
      do 7 i = 1,totlrows
        if (code .eq. 'y') then
          write (7,300), (dsgnmtrx(i,j), j = 1,paramtrs)
        else
          write (8,350), (dsgnmtrx(i,j), j = 1,totlcols)
        endif
    7 continue
      if (code .eq. 'y') then
        close(7)
      else
        close(8)
      endif
  300 format (' ',16(F4.1,1X))
  325 format (' ',2(I2,1X))
  350 format (' ',64(F7.2,1X))
  375 format (' ',I2)
  400 format (' ',3(1X,F7.2),1X,I1)
      return
      end
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                             c
      Subroutine SetUp
c                                                                             c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:
c
c
c       Variables:
```

```
c
c
      character*1 dsgncode(2:5,16,5)
      integer*2 dsgnmbrs(3:16,24,5)
      common /aspects/dsgncode,dsgnmbrs
      integer*2 j,k
      integer*2 dsgnmbr3(3,2)
      integer*2 dsgnmbr4(6,2)
      integer*2 dsgnmbr5(10,2)
      integer*2 dsgnmbr6(6,3)
      integer*2 dsgnmbr7(7,3)
      integer*2 dsgnmbr9(15,3)
      integer*2 dsgnmbra(10,4)
      integer*2 dsgnmbrb(11,5)
      integer*2 dsgnmbrc(12,4)
      integer*2 dsgnmbrg(24,4)
      character*1 dsgncde2(4,2)
      character*1 dsgncde3(8,3)
      character*1 dsgncde4(16,4)
      character*1 dsgn5frc(16,5)
      data dsgnmbr3/ 1, 1, 2,
     &               2, 3, 3/
      data dsgnmbr4/ 1, 3, 1, 2, 1, 2,
     &               2, 4, 4, 3, 3, 4/
      data dsgnmbr5/ 1, 3, 2, 1, 4, 2, 1, 3, 1, 2,
     &               2, 4, 5, 3, 5, 3, 4, 5, 5, 4/
      data dsgnmbr6/ 1, 2, 3, 1, 2, 1,
     &               2, 3, 4, 4, 5, 3,
     &               4, 5, 6, 5, 6, 6/
      data dsgnmbr7/ 4, 1, 2, 1, 3, 1, 2,
     &               5, 6, 5, 2, 4, 3, 3,
     &               6, 7, 7, 4, 7, 5, 6/
      data dsgnmbr9/ 1, 2, 3, 1, 4, 7, 1, 3, 2, 1, 2, 3, 1, 2, 3,
     &               4, 5, 6, 2, 5, 8, 5, 4, 6, 6, 4, 5, 4, 5, 6,
     &               7, 8, 9, 3, 6, 9, 9, 8, 7, 8, 9, 7, 7, 8, 9/
      data dsgnmbra/ 2, 1, 2, 2, 1, 3, 1, 3, 1, 4,
     &               6, 2, 3, 4, 8, 4, 4, 5, 3, 5,
     &               7, 5, 7, 6, 9, 5, 7, 7, 6, 6,
     &              10,10, 8, 9,10,10, 8, 9, 9, 8/
      data dsgnmbrb/ 3, 1, 2, 1, 1, 1, 2, 3, 4, 1, 2,
     &               7, 4, 5, 3, 2, 2, 3, 4, 5, 5, 6,
     &               8, 8, 9, 6, 4, 3, 4, 5, 6, 6, 7,
     &               9, 9,10,10, 7, 5, 6, 7, 8, 7, 8,
     &              11,10,11,11,11, 8, 9,10,11, 9,10/
      data dsgnmbrc/ 1, 2, 3, 4, 5, 6, 1, 2, 1, 2, 3, 1,
     &               2, 3, 4, 5, 6, 7, 7, 8, 3, 4, 5, 4,
     &               5, 6, 7, 8, 9,10, 8, 9, 9,10,11, 6,
     &               7, 8, 9,10,11,12,11,12,10,11,12,12/
      data dsgnmbrg/ 1, 3, 5, 7, 2, 1, 6, 8, 2, 4, 1, 3, 3, 1, 2, 4,
     &               1, 2, 5, 6, 4, 3, 1, 2,
     &               2, 4,10,12, 3, 4,11, 9, 5, 7, 9,11, 6, 5,10, 9,
     &               3, 4, 7, 8, 6, 5, 7, 8,
     &               6, 8,13,15, 7, 5,14,13, 6, 8,10,12, 7, 8,11,12,
```

```
     &              13,14, 9,10,10, 9,11,12,
     &               9,11,14,16,10,12,15,16,13,15,14,16,14,16,15,13,
     &              15,16,11,12,16,15,13,14/
      data dsgncde2/'-','+','-','+',
     &              '-','-','+','+'/
      data dsgncde3/'-','+','-','+','-','+','-','+',
     &              '-','-','+','+','-','-','+','+',
     &              '-','-','-','-','+','+','+','+'/
      data dsgncde4/'+','+','+','+','+','+','-','-','-','-','-','-','-',
     &              '-','+','+',
     &              '+','+','+','+','-','-','-','+','+','+','+','-','-',
     &              '-','-','-',
     &              '+','+','-','-','-','+','+','+','+','-','-','-','-',
     &              '+','+','-',
     &              '+','-','-','+','+','+','+','+','-','-','+','+','-',
     &              '-','-','-'/
c      data dsgncde4/'-','+','-','+','-','+','-','+','-','+','-','+','-',
c     &              '+','-','+',
c     &              '-','-','+','+','-','-','+','+','-','-','+','+','-',
c     &              '-','+','+',
c     &              '-','-','-','-','+','+','+','+','-','-','-','-','+',
c     &              '+','+','+',
c     &              '-','-','-','-','-','-','-','-','+','+','+','+','+',
c     &              '+','+','+'/
      data dsgn5frc/'-','+','-','+','-','+','-','+','-','+','-','+','-',
     &              '+','-','+',
     &              '-','-','+','+','-','-','+','+','-','-','+','+','-',
     &              '-','+','+',
     &              '-','-','-','-','+','+','+','+','-','-','-','-','+',
     &              '+','+','+',
     &              '-','-','-','-','-','-','-','-','+','+','+','+','+',
     &              '+','+','+',
     &              '+','-','-','+','-','+','+','-','-','+','+','-','+',
     &              '-','-','+'/
c
c
c       Begin:
c
c
      data dsgnmbrs/1680*0/
      data dsgncode/320*' '/
      do 1 j = 1,3
        do 2 k = 1,2
          dsgnmbrs(3,j,k) = dsgnmbr3(j,k)
   2    continue
   1  continue
      do 3 j = 1,6
        do 4 k = 1,2
          dsgnmbrs(4,j,k) = dsgnmbr4(j,k)
   4    continue
   3  continue
      do 5 j = 1,10
        do 6 k = 1,2
```

```
          dsgnmbrs(5,j,k) = dsgnmbr5(j,k)
 6      continue
 5    continue
      do 7 j = 1,6
        do 8 k = 1,3
          dsgnmbrs(6,j,k) = dsgnmbr6(j,k)
 8      continue
 7    continue
      do 9 j = 1,7
        do 10 k = 1,3
          dsgnmbrs(7,j,k) = dsgnmbr7(j,k)
10      continue
 9    continue
      do 11 j = 1,15
        do 12 k = 1,3
          dsgnmbrs(9,j,k) = dsgnmbr9(j,k)
12      continue
11    continue
      do 13 j = 1,10
        do 14 k = 1,4
          dsgnmbrs(10,j,k) = dsgnmbra(j,k)
14      continue
13    continue
      do 15 j = 1,11
        do 16 k = 1,5
          dsgnmbrs(11,j,k) = dsgnmbrb(j,k)
16      continue
15    continue
      do 17 j = 1,12
        do 18 k = 1,4
          dsgnmbrs(12,j,k) = dsgnmbrc(j,k)
18      continue
17    continue
      do 19 j = 1,24
        do 20 k = 1,4
          dsgnmbrs(16,j,k) = dsgnmbrg(j,k)
20      continue
19    continue
      do 21 j = 1,4
        do 22 k = 1,2
          dsgncode(2,j,k) = dsgncde2(j,k)
22      continue
21    continue
      do 23 j = 1,8
        do 24 k = 1,3
          dsgncode(3,j,k) = dsgncde3(j,k)
24      continue
23    continue
      do 25 j = 1,16
        do 26 k = 1,4
          dsgncode(4,j,k) = dsgncde4(j,k)
26      continue
25    continue
```

```
      do 27 j = 1,16
        do 28 k = 1,5
          dsgncode(5,j,k) = dsgn5frc(j,k)
  28    continue
  27  continue
      return
      end
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                             c
      subroutine Design3(code)
c                                                                             c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c     Purpose:
c
c
c     Variables:
c
c
      real dsgnmtrx(385,64)
      common /dsgnmtrx/dsgnmtrx
      character*1 dsgncode(2:5,16,5)
      integer*2 dsgnmbrs(3:16,24,5)
      common /aspects/dsgncode,dsgnmbrs
      integer*2 paramtrs,controls(16)
      double precision low(16),mid(16),high(16)
      common /uncode/paramtrs,low,mid,high,controls
      integer*2 features(3:16,3)
      common /features/features
      real desgnvar(1,16)
      integer*2 i,j,k,col,row,rowlines,lines,s,totlrows
      character*1 plus,minus,code
      data plus/'+'/minus/'-'/
      data dsgnmtrx/24640*0/
c
c
c       Begin:
c
c
      lines = features(paramtrs,1)
      rowlines = features(paramtrs,2)
      s = features(paramtrs,3)
      do 4 i = 1,lines
        do 5 j = 1,rowlines
          row = j + ((i - 1) * rowlines)
          col = 1
          do 6 k = 1,paramtrs
            if (col .le. s) then
              if (k .eq. dsgnmbrs(paramtrs,i,col)) then
```

```
                  if (dsgncode(s,j,col) .eq. plus) then
                    dsgnmtrx(row,k) = 1
                  else
                    if (dsgncode(s,j,col) .eq. minus) then
                      dsgnmtrx(row,k) = -1
                    else
                      print *, 'Error at row ',row,' and column ',k
                    endif
                  endif
                col = col + 1
                endif
              endif
    6       continue
    5     continue
    4   continue
        if (code .eq. 'n') then
          totlrows = row + 1
          do 1 i = 1,totlrows
            col = 0
            do 7 j = 1,paramtrs
              desgnvar(i,j) = dsgnmtrx(i,j)
    7       continue
            do 2 j = 1,paramtrs
              do 3 k = 1,controls(j)
                col = col + 1
                if (desgnvar(i,j) .eq. 1.0) then
                  dsgnmtrx(i,col) = high(j)
                else
                  if (desgnvar(i,j) .eq. 0.0) then
                    dsgnmtrx(i,col) = mid(j)
                  else
                    dsgnmtrx(i,col) = low(j)
                  endif
                endif
    3         continue
    2       continue
    1     continue
        endif
        return
        end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                            c
       Subroutine Rgrssion
c                                                                            c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c      Purpose:
c
c
```

```
c       Variables:
c
c
      integer*2 selction(4)
      common /choices/selction
      integer*2 response(5)
      common /respnses/response
      integer*2 i,j,paramtrs,totlrows,totlcols,objctive,process,dummyarg
      logical filesok
c
c       Begin:
c
c     The forthcoming Fortran code performs, respectively, the following steps:
c
c        1)  Obtain input for this subroutine;
c        2)  Initialize those variables and matrices that require it;
c        3)  Compute the 'Xprime' matrix;
c        4)  Compute the 'XprimeX' matrix;
c        5)  Compute the 'XprXInvr' matrix;
c        6)  Compute the 'Yprime' matrix;
c        7)  Compute the 'YprimeY' matrix;
c        8)  Compute the 'XprimeY' matrix;
c        9)  Compute the 'B' matrix;
c       10)  Compute the 'Bprime' matrix;
c       11)  Compute the 'BprXprY' matrix;
c       12)  Compute the 'Onesprme' matrix;
c       13)  Compute the 'YprmOnes' matrix;
c       14)  Compute the 'OnesprmY' matrix; and
c       15)  Compute the 'YprllprY' matrix.
c
      dummyarg = 0
      filesok = .true.
      call RgrsInpt(paramtrs,totlrows,process,filesok)
      if (filesok) then
        call ConstVls(paramtrs,totlrows,totlcols)
        do 2 i = 1,process
          objctive = response(i)
          do 1 j = 1,2
            call RgrsInit(paramtrs,totlrows,totlcols,objctive,
     &                    selction(j))
            if (j .eq. 1) then
              call orthognl(totlrows,paramtrs)
            endif
            call Trnspose(totlrows,totlcols,dummyarg,selction(1))
            call MtrxMply(totlcols,totlrows,totlcols,dummyarg,
     &                    selction(1))
            call RgrsInvr(totlcols,paramtrs,selction(j))
            if (j .eq. 2) then
              call Trnspose(totlrows,dummyarg,objctive,selction(2))
              call MtrxMply(dummyarg,totlrows,dummyarg,objctive,
     &                      selction(2))
            endif
            call MtrxMply(totlcols,totlrows,totlcols,objctive,
```

```
     &                    selction(3))
            call Trnspose(totlcols,dummyarg,objctive,selction(3))
            if (j .eq. 1) then
              call CmpStats(totlrows,totlcols,objctive,selction(1))
            else
              call MtrxMply(totlrows,totlcols,totlrows,objctive,
     &                      selction(4))
              call CmpStats(totlrows,totlcols,objctive,selction(2))
            endif
    1     continue
    2   continue
        call RgrsOput(totlcols,totlrows,process,response)
        call ForDcode(paramtrs,totlcols,process,response)
      endif
c
c     Time to exit:
c
      return
      end
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
      Subroutine RgrsInpt(paramtrs,totlrows,process,filesok)
c                                                                       c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  To input from external sources the required data necessary
c                 to execute the regression package.
c
c
c       Variables:
c
c
      double precision TheDesign(385,16)
      common /TheDesign/TheDesign
      character*8 parmnmes(16),blanks8
      common /TheNames/parmnmes,blanks8
      integer*2 paramtrs,totlrows,objctves,process
      logical filesok
      character readcr*1,filename*8,fileindx*4
      integer*2 i,j,lines,rowlines,whichone
c
c     Begin:
c
c     Obtain the number of parameters and their names.
c
      blanks8 = '        '
      whichone = 1
      open (unit=1,err=8,file='rgrssion.in',status='old')
      whichone = 2
```

```
      read (1,11,err=8) paramtrs
      do 1 i = 1,paramtrs
        parmnmes(i) = blanks8
        read (1,12) parmnmes(i)
    1 continue
      close (1)
c
c     Obtain the matrix design.
c
      whichone = 1
      open (unit=2,err=9,file='dsgnmtrx.in',status='old')
      whichone = 2
      read (2,13,err=9) lines,rowlines
      totlrows = (lines*rowlines) + 1
      do 5 i = 1,totlrows
        read (2,14,err=9) (TheDesign(i,j), j = 1,paramtrs)
    5 continue
      close (2)
c
c     Go get the responses:
c
      call HowMnyYs(objctves,process)
      call GetRspns(objctves,process,totlrows,filesok)
c
c     Get out of here if no errors:
c
      go to 15
c
c     Error code:
c
    8 if (whichone .eq. 2) then
        close (1)
      endif
      filename = 'RGRSSION'
      fileindx = '.IN '
      go to 10
    9 if (whichone .eq. 2) then
        close (2)
      endif
      filename = 'DSGNMTRX'
      fileindx = '.IN '
   10 call Filerror(filename,fileindx,whichone)
      filesok = .false.
      go to 15
c
c     Format statements:
c
   11 format (1X,I2)
   12 format (1X,A8)
   13 format (2(1X,I2))
   14 format (16(1X,F4.1))
c
c     Another subroutine finished:
```

```
c
  15  return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine HowMnyYs(objctves,process)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:
c
c
c       Variables:
c
c
      integer*2 objctves,process,objctlmt
      character*1 readcr
c
c     Begin:
c
c     Receive the total number of objectives from the user:
c
      objctlmt = 5
    1 call ScrlScrn(11)
      print 5, objctlmt
      call ScrlScrn(11)
      read (*,6,err=2) objctves
      if (objctves .lt. 1 .or. objctves .gt. objctlmt) then
    2   call ScrlScrn(11)
        print 7, objctlmt
        call ScrlScrn(11)
        accept 8, readcr
        go to 1
      endif
c
c     Receive the number of objectives the user wishes to process:
c
    3 call ScrlScrn(11)
      print 9
      call ScrlScrn(11)
      read (*,6,err=4) process
      if (process .lt. 1 .or. process .gt. objctves) then
    4   call ScrlScrn(11)
        print 7, objctves
        call ScrlScrn(11)
        accept 8, readcr
        go to 3
      endif
c
```

```
c     Format Statements:
c
    5  format (' ','How many response objectives (total) are there?  ',
     &        '(Max of ',I1,'):')
    6  format (I)
    7  format (' ','Value has to be an integer between 1 and ',I1,'!!',
     &        /,' Press "return" to reenter value:')
    8  format (a)
    9  format (' ','How many of the total response objectives do you ',
     &'wish to process?')
c
c     Through:
c
      return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine GetRspns(objctves,process,totlrows,filesok)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:
c
c
c       Variables:
c
c
      double precision Y(385,5)
      common /rspnsmtc/Y
      integer*2 response(5)
      common /respnses/response
      integer*2 objctves,process,totlrows
      logical filesok
      integer*2 i,j,k,whichone
      character ending(5)*2,readcr*1,filename*8,fileindx*4
      data ending/'st','nd','rd',2*'th'/
c
c     Begin:
c
c     Read in the responses and put into matrix 'Y'.
c
      whichone = 1
      open (unit=3,err=8,file='respnses.in',status='old')
      whichone = 2
      if (objctves .eq. process) then
c
c       If user wishes to process all responses then:
c
        do 1 i = 1,totlrows
```

```
          read (3,14,err=8), (Y(i,j), j = 1,objctves)
    1   continue
        do 2 i = 1,5
          response(i) = i
    2   continue
      else
c
c       Else, find out which responses the user wants to process.
c
        do 7 j = 1,process
    3     call ScrlScrn(11)
          print 9, j,ending(j)
          call ScrlScrn(11)
          read (*,10,err=4) response(j)
          if (response(j) .lt. 1 .or. response(j) .gt. objctves) then
c
c           If user's input is not an integer and not within the bounds,
c           make him/her reenter.
c
    4       call ScrlScrn(11)
            print 11, objctves
            call ScrlScrn(11)
            accept 12, readcr
            go to 3
          endif
          do 5 k = 1,j
            if (k .ne. j) then
              if (response(k) .eq. response(j)) then
c
c               If the user tries to process a response twice, send him/her
c               back to the beginning to reenter his/her answer.
c
                call ScrlScrn(10)
                print 13, response(j),k,ending(k)
                call ScrlScrn(11)
                accept 12, readcr
                go to 3
              endif
            endif
    5     continue
c
c         Read in the response values the user wants to process:
c
          k = response(j)
          rewind (3)
          do 6 i = 1,totlrows
            if (k .eq. 1) then
              read (3,14,err=8), Y(i,k)
            else
              if (k .eq. 2) then
                read (3,15,err=8), Y(i,k)
              else
                if (k .eq. 3) then
```

```
                read (3,16,err=8), Y(i,k)
              else
                if (k .eq. 4) then
                  read (3,17,err=8), Y(i,k)
                else
                  read (3,18,err=8), Y(i,k)
                endif
              endif
            endif
          endif
    6     continue
    7   continue
      endif
      close (3)
      go to 19
c
c     Error code:
c
    8 if (whichone .eq. 2) then
        close (3)
      endif
      filename = 'RESPNSES'
      fileindx = '.IN '
      call Filerror(filename,fileindx,whichone)
      filesok = .false.
      go to 19
c
c     Format Statements:
c
    9 format (' ','What is the number of the ',I1,A2,' response ',
     &        'objective?')
   10 format (I)
   11 format (' ','Value has to be an integer between 1 and ',I1,'!!',
     &       /,' Press "return" to reenter value:')
   12 format (a)
   13 format (' ','You have already entered the value ',I1,' for the ',
     &I1,A2,' response objective!!',/,' Press "return" to reenter:')
   14 format (5(1X,F6.2))
   15 format (8X,F6.2)
   16 format (15X,F6.2)
   17 format (22X,F6.2)
   18 format (29X,F6.2)
c
c     Another masterpiece!!
c
   19 return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                      c
      Subroutine ConstVls(paramtrs,totlrows,totlcols)
```

```
c                                                                          c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  To initialize and setup those variables and matrices whose
c                 values never change thoughout the program.
c
c
c       Variables:
c
c
      character varname1(153)*8,varname2(153)*8,times(153)*3
      common /varnames/varname1,varname2,times
      character*8 parmnmes(16),blanks8
      common /TheNames/parmnmes,blanks8
      integer*2 selction(4)
      common /choices/selction
      double precision Ones,Onesprme
      common /onesmtcs/Ones,Onesprme
      integer*2 paramtrs,totlrows,totlcols
      integer*2 i,j,k,l,lincol,col,nextprms
c
c     Begin:
c
c     Initialize the variable names.
c
      lincol = paramtrs + 1
      do 2 j = 1,paramtrs
        col = lincol + j
        varname1(col) = parmnmes(j)
        varname2(col) = blanks8
        times(col) = '   '
    2 continue
      do 4 j = 1,paramtrs
        do 3 k = j,paramtrs
          if (k .eq. j) then
            l = j + 1
            varname1(l) = parmnmes(j)
            varname2(l) = parmnmes(k)
            times(l) = ' * '
          else
            col = col + 1
            varname1(col) = parmnmes(j)
            varname2(col) = parmnmes(k)
            times(col) = ' * '
          endif
    3   continue
    4 continue
      varname1(1) = 'INTERCEP'
      varname2(1) = blanks8
      times(1) = 'T  '
c
c       Compute the total columns in the Regression matrix:
```

```
c
      totlcols = 1 + paramtrs
      do 6 i = 1,paramtrs
        nextprms = paramtrs - (i - 1)
        totlcols = totlcols + nextprms
   6  continue
c
c     Set the 'selction' array:
c
      do 7 i = 1,4
        selction(i) = i
   7  continue
c
c     Initialize matrix 'Ones':
c
      ones = 1.0
c
c     Create the transpose of matrix 'Ones', i.e., 'Onesprme'.
c
      Onesprme = Ones
c
c     Another subroutine finished:
c
      return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine RgrsInit(paramtrs,totlrows,totlcols,objctive,choice)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  Initialize and compute the variables and matrices necessary
c                 for regression computations.
c
c
c       Variables:
c
c
      double precision TheDesign(385,16)
      common /TheDesign/TheDesign
      double precision X(385,153)
      common /theXmtrx/X
      double precision XprXInvr(153,153)
      common /XprXInvr/XprXInvr
      double precision SaveX(385,153),SaveInvr(153,153)
      common /SaveMtcs/SaveX,SaveInvr
      integer*2 totldltd(153)
      common /totldltd/totldltd
      integer*2 paramtrs,totlrows,totlcols,objctive,choice
```

```
      integer*2 i,j,k,l,m,lincol,col,nextprms
c
c     Begin:
c
c     First, initialize the 'totldltd' array:
c
      go to (1,2),choice
    1 do 11 i = 1,totlcols
        totldltd(i) = 0
   11 continue
c
c     Second, compute the regression matrix ('X') by expanding 'TheDesign'
c     matrix.
c
      lincol = paramtrs + 1
      do 15 i = 1,totlrows
        X(i,1) = 1.0
        do 12 j = 1,paramtrs
          col = lincol + j
          X(i,col) = TheDesign(i,j)
   12   continue
        l = 0
        do 14 j = 1,paramtrs
          do 13 k = j,paramtrs
            if (k .eq. j) then
              m = j + 1
              X(i,m) = TheDesign(i,j) * TheDesign(i,k)
            else
              l = l + 1
              col = l + (2 * paramtrs) + 1
              X(i,col) = TheDesign(i,j) * TheDesign(i,k)
            endif
   13     continue
   14   continue
   15 continue
c
c     Check if user wants to modify the regression matrix.
c
      call Stepwise(totlrows,totlcols,objctive)
c
c     Save the 'X' matrix for second init.
c
      do 17 i = 1,totlrows
        do 16 j = 1,totlcols
          SaveX(i,j) = X(i,j)
   16   continue
   17 continue
c
c     Initialize the XprXInvr matrix by setting the diagonal elements to 1.0:
c
      do 19 i = 1,totlcols
        do 18 j = 1,totlcols
          if (i .eq. j) then
```

```
            XprXInvr(i,j) = 1.0
          else
            XprXInvr(i,j) = 0.0
          endif
   18   continue
   19 continue
c
c     Save the 'XprXInvr' matrix.
c
      do 22 i = 1,totlcols
        do 21 j = 1,totlcols
          SaveInvr(i,j) = XprXInvr(i,j)
   21   continue
   22 continue
      return
c
c     The second init:  First, reinit the 'X' matrix.
c
    2 do 40 i = 1,totlrows
        do 41 j = 1,totlcols
          X(i,j) = SaveX(i,j)
   41   continue
   40 continue
c
c     Second, reinit the 'XprXInvr' matrix.
c
      do 43 i = 1,totlcols
        do 42 j = 1,totlcols
          XprXInvr(i,j) = SaveInvr(i,j)
   42   continue
   43 continue
      return
c
c     Bye, Bye again:
c
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine Stepwise(totlrows,totlcols,objctive)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  Initialize and compute the variables and matrices necessary
c                 for regression computations.
c
c
c       Variables:
c
c
```

```
      integer*2 totldlts
      common /totldlts/totldlts
      integer*2 dletions
      common /dletions/dletions
      character*2 dpndntnm(5)
      common /dpndntnm/dpndntnm
      integer*2 totlrows,totlcols,objctive
      integer*2 whchline
      character*1 modify,goodchar,deleted,repeat
      data dpndntnm/'Y1','Y2','Y3','Y4','Y5'/
c
c       Begin:
c
      dletions = 0
      repeat = 'y'
      whchline = 1
   1  if (repeat .eq. 'y' .or. repeat .eq. 'Y') then
        call ScrlScrn(11)
        if (whchline .eq. 1) then
          print 3, dpndntnm(objctive)
        else
          print 4, dpndntnm(objctive)
        endif
        call ScrlScrn(11)
        accept 2, modify
        call ChckChar(modify,goodchar)
        if (goodchar .ne. 'y') then
          call BadChar
          go to 1
        endif
        whchline = whchline + 1
        if (modify .eq. 'y' .or. modify .eq. 'Y') then
          call RdceMtrx(totlrows,totlcols,objctive,deleted)
          if (deleted .eq. 'y') then
            dletions = dletions + 1
          endif
        else
          repeat = 'n'
        endif
        go to 1
      endif
      totldlts = dletions
c
c     Format statements:
c
   2    format (a)
   3  format (' ','Do you wish to delete a variable in response ',
     &'objective ',A2,'?  (y/n)')
   4  format (' ','Do you wish to delete another variable in response',
     &' objective ',A2,'?  (y/n)')
c
c     Adios:
c
```

```
      return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine RdceMtrx(totlrows,totlcols,objctive,deleted)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  Initialize and compute the variables and matrices necessary
c                 for regression computations.
c
c
c       Variables:
c
c
      double precision X(385,153)
      common /theXmtrx/X
      integer*2 totlrows,totlcols,objctive
      integer*2 i,col
      character*1 okay,readcr,deleted
c
c     Begin:
c
   1  call ShowVars(totlcols,objctive)
      read (*,7,err=2) col
      call ChckNmbr(col,totlcols,okay)
      if (okay .eq. 'n') then
   2    call ScrlScrn(11)
        print 5, totlcols
        call ScrlScrn(10)
        accept 6, readcr
        go to 1
      else
        if (col .eq. 0) then
          deleted = 'n'
        else
          do 4 i = 1,totlrows
            X(i,col) = 0.0
   4      continue
          deleted = 'y'
        endif
      endif
c
c     Format statements:
c
   5  format (' ','Reminder, value has to be an integer between 2 ',
     &'and ',I3,', or zero.',/,' Press "return" to try again:')
   6  format (a)
   7  format (I3)
```

```
c
c     Hope it was good for you; it was good for me!!
c
      return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine ShowVars(totlcols,objctive)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  Initialize and compute the variables and matrices necessary
c                 for regression computations.
c
c
c       Variables:
c
c
      character varname1(153)*8,varname2(153)*8,times(153)*3
      common /varnames/varname1,varname2,times
      character*2 dpndntnm(5)
      common /dpndntnm/dpndntnm
      integer*2 totlcols,objctive
      integer*2 i,j,k,lnstoscl,linestkn
      character*1 readcr
c
c     Begin:
c
      linestkn = 0
      do 1 i = 1,totlcols,3
        if ((totlcols - i) .ge. 2) then
          linestkn = linestkn + 1
          j = i + 1
          k = i + 2
          print 2, i,varname1(i),times(i),varname2(i),
     &              j,varname1(j),times(j),varname2(j),
     &              k,varname1(k),times(k),varname2(k)
          if (linestkn .eq. 20) then
            linestkn = 0
            call ScrlScrn(1)
            print *, 'Press "return" to continue'
            call ScrlScrn(1)
            accept 3, readcr
          endif
        else
          linestkn = linestkn + 1
          if ((totlcols - i) .eq. 1) then
            j = i + 1
            print 2, i,varname1(i),times(i),varname2(i),
```

```
     &               j,varname1(j),times(j),varname2(j)
          else
            print 2, i,varname1(i),times(i),varname2(i)
          endif
        endif
    1  continue
       lnstoscl = 20 - linestkn
       call ScrlScrn(lnstoscl)
       print 4, dpndntnm(objctive),totlcols
       call ScrlScrn(1)
c
c      Format Statements:
c
    2  format (' ',3(2X,I3,':',1X,A8,A3,A8))
    3  format (a)
    4  format (' ','For ',A2,', which variable, between 2 and ',I3,
     &', do you wish to delete?',/,' Enter zero (0) for none:')
c
c      No more "vars" to show:
c
       return
       end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                      c
       Subroutine ChckNmbr(col,totlcols,okay)
c                                                                      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c        Purpose:  Initialize and compute the variables and matrices necessary
c                  for regression computations.
c
c
c        Variables:
c
c
       integer*2 dletions
       common /dletions/dletions
       integer*2 totldltd(153)
       common /totldltd/totldltd
       integer*2 i,col,totlcols
       character*1 okay,readcr,deleted,cutalrdy,finished
c
c      Begin:
c
       if (col .ge. 2 .and. col .le. totlcols) then
         okay = 'y'
       else
         if (col .eq. 0) then
           okay = 'y'
```

```
          else
            okay = 'n'
          endif
        endif
        if (okay .eq. 'y') then
          cutalrdy = 'n'
          finished = 'n'
          i = 0
    1     if (cutalrdy .eq. 'n' .and. finished .eq. 'n') then
            i = i + 1
            if (i .le. dletions) then
              if (col .eq. totldltd(i)) then
                cutalrdy = 'y'
              endif
            else
              finished = 'y'
            endif
          go to 1
          endif
          if (cutalrdy .eq. 'y') then
            call ScrlScrn(11)
            write (*,2) col
            call ScrlScrn(10)
            accept 3, readcr
            okay = 'n'
          else
            totldltd(i) = col
          endif
        endif
    2   format (' ','The variable number you inputted, number ',I3,
       &', has already been deleted!!',/,' Press "return" to continue:')
    3   format (a)
        return
        end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                      c
      Subroutine Orthognl(rows,paramtrs)
c                                                                      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c         Purpose:  To create an orthogonal matrix out or the regression matrix,
c                   i.e., 'X', by using the Gramm-Schmidt process.
c
c
c         Variables:
c
c
      double precision X(385,153)
      common /theXmtrx/X
```

```
      integer*2 rows,paramtrs
      double precision PfctMtrx(385,153),Divisors(17)
      double precision SSbyCols,xmltplir
      integer*2 columns,xcol,vcol,row
c
c       Begin:
c
      columns = paramtrs + 1
      do 5 xcol = 1,columns
        do 1 row = 1,rows
          PfctMtrx(row,xcol) = X(row,xcol)
   1    continue
        do 4 vcol = 1,xcol
          SSbyCols = 0.0
          do 2 row = 1,rows
            SSbyCols = SSbyCols+(PfctMtrx(row,xcol)*PfctMtrx(row,vcol))
   2      continue
          if (vcol .lt. xcol) then
            if (Divisors(vcol) .gt. 0) then
              xmltplir = SSbyCols / Divisors(vcol)
              do 3 row = 1,rows
                PfctMtrx(row,xcol) = PfctMtrx(row,xcol) -
     &                               (xmltplir * PfctMtrx(row,vcol))
   3          continue
            endif
          else
            Divisors(xcol) = SSbyCols
          endif
   4    continue
   5  continue
      do 7 xcol = 1,columns
        do 6 row = 1,rows
          X(row,xcol) = PfctMtrx(row,xcol)
   6    continue
   7  continue
c
c     Another one bites the dust!!
c
      return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine Trnspose(row,column,objctive,choice)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  Take the transpose of a matrix.  Control is managed through
c                 the variable 'choice'.
c
```

```
c
c       Variables:
c
c
      double precision B(153,5)
      common /theBmtrx/B
      double precision Xprime(153,385),Yprime(5,385)
      common /trnspses/Xprime,Yprime
      double precision X(385,153)
      common /theXmtrx/X
      double precision Y(385,5)
      common /rspnsmtc/Y
      double precision Bprime(5,153)
      common /Bprime/Bprime
      integer*2 row,column,objctive,choice
      integer*2 i,j
c
c       Begin:
c
      go to (1,2,3),choice
c
c     Create the transpose of matrix 'X', i.e., 'Xprime'.
c
    1 do 11 i = 1,row
        do 10 j = 1,column
          Xprime(j,i) = X(i,j)
   10   continue
   11 continue
      return
c
c     Create the transpose of matrix 'Y', i.e., 'Yprime'.
c
    2 j = objctive
      do 13 i = 1,row
        Yprime(j,i) = Y(i,j)
   13 continue
      return
c
c     Create the transpose of matrix 'B', i.e., 'Bprime'.
c
    3 j = objctive
      do 15 i = 1,row
        Bprime(j,i) = B(i,j)
   15 continue
c
c     That's all folks!!
c
      return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
c                                                                           c
      Subroutine MtrxMply(LeftSide,Inside,RghtSide,objctive,choice)
c                                                                           c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  Multiply two matrices together.  Control is achieved through
c                 the variable 'choice'.
c
c       Variables:
c
c
      double precision Y(385,5)
      common /rspnsmtc/Y
      double precision X(385,153)
      common /theXmtrx/X
      double precision XprXInvr(153,153)
      common /XprXInvr/XprXInvr
      double precision Ones,Onesprme
      common /onesmtcs/Ones,Onesprme
      double precision Xprime(153,385),Yprime(5,385)
      common /trnspses/Xprime,Yprime
      double precision Bprime(5,153)
      common /Bprime/Bprime
      double precision B(153,5)
      common /theBmtrx/B
      double precision XprimeX(153,153)
      common /XprimeX/XprimeX
      double precision YprimeY,BprXprY,XprimeY(153,5),Ypr1lprY
      common /statmtcs/YprimeY,BprXprY,XprimeY,Ypr1lprY
      double precision XtimesB(385,5)
      common /XtimesB/XtimesB
      double precision YprmOnes,OnesprmY
      integer*2 LeftSide,Inside,RghtSide,objctive,choice
      integer*2 i,j,k
c
c       Begin:
c
      j = objctive
      go to (1,2,3,4),choice
c
c       Multiply matrices 'Xprime' and 'X' and put their product in
c       matrix 'XprimeX'.
c
    1 do 23 i = 1,LeftSide
        do 22 j = 1,RghtSide
          XprimeX(i,j) = 0.0
          do 21 k = 1,Inside
            XprimeX(i,j) = XprimeX(i,j) + (Xprime(i,k) * X(k,j))
   21     continue
   22   continue
   23 continue
      return
```

```
c
c       Multiply matrices 'Yprime' and 'Y' and put their product in
c       matrix 'YprimeY'.
c
    2 YprimeY = 0.0
      do 24 k = 1,Inside
        YprimeY = YprimeY + (Yprime(j,k) * Y(k,j))
   24 continue
      return
c
c       Multiply matrices 'Xprime' and 'Y' and put their product in
c       matrix 'XprimeY'.
c
    3 do 29 i = 1,LeftSide
        XprimeY(i,j) = 0.0
        do 27 k = 1,Inside
          XprimeY(i,j) = XprimeY(i,j) + (Xprime(i,k) * Y(k,j))
   27   continue
   29 continue
c
c       Multiply matrices 'XprXInvr' and 'Xprime' and put their product in
c       matrix 'B'.
c
      do 32 i = 1,LeftSide
        B(i,j) = 0.0
        do 30 k = 1,RghtSide
          B(i,j) = B(i,j) + (XprXInvr(i,k) * XprimeY(k,j))
   30   continue
   32 continue
      return
c
c       Multiply matrices 'Bprime' and 'XprimeY' and put their product in
c       matrix 'BprXprY'.
c
    4 BprXprY = 0.0
        do 33 k = 1,Inside
          BprXprY = BprXprY + (Bprime(j,k) * XprimeY(k,j))
   33   continue
c
c     Multiply matrices 'Yprime' and 'Ones' and put their product in
c     matrix 'YprmOnes'.
c
      YprmOnes = 0.0
      do 36 k = 1,RghtSide
        YprmOnes = YprmOnes + (Yprime(j,k) * Ones)
   36 continue
c
c       Multiply matrices 'Onesprme' and 'Y' and put their product in
c       matrix 'OnesprmY'.
c
      OnesprmY = 0.0
      do 39 k = 1,LeftSide
        OnesprmY = OnesprmY + (Onesprme * Y(k,j))
```

```
   39  continue
c
c        Multiply matrices 'YprmOnes' and 'OnesprmY' and put their product in
c        matrix 'Ypr1lprY'.
c
       Ypr1lprY = 0.0
       Ypr1lprY = Ypr1lprY + (YprmOnes * OnesprmY)
c
c        Multiply matrices 'X' and 'B' and put their product in matrix
c        'XtimesB'.
c
       do 41 i = 1,leftside
         XtimesB(i,j) = 0.0
         do 40 k = 1,inside
           XtimesB(i,j) = XtimesB(i,j) + (X(i,k) * B(k,j))
    40   continue
    41 continue
       return
c
c      Finished!!  Through!!  Capice!!
c
       end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
       Subroutine RgrsInvr(totlcols,paramtrs,choice)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c        Purpose:  Take inverse of XprimeX matrix.
c
c
c        Variables:
c
c
       double precision XprXInvr(153,153)
       common /XprXinvr/XprXInvr
       double precision XprimeX(153,153)
       common /XprimeX/XprimeX
       double precision rwmplier
       integer*2 i,j,k,totlcols,columns,paramtrs,choice
c
c        Begin:
c
       go to (1,2),choice
c
c      Take the inverse of an orthogonal matrix:
c
     1 do 10 i = 1,totlcols
         if (XprimeX(i,i) .eq. 0) then
```

```
          go to 10
        endif
        XprXInvr(i,i) = XprXInvr(i,i) / XprimeX(i,i)
  10  continue
      return
c
c     Take the inverse of a non-orthogonal matrix:
c
   2  columns = paramtrs + 1
      do 13 i = 1,columns
        if (XprimeX(i,i) .eq. 0) then
          go to 13
        endif
c
ccc     Find the non-zero element within the column of the iith element:
c
        do 12 j = 1,columns
          if (i .eq. j) then
            go to 12
          else
            if (XprimeX(j,i) .eq. 1.0E-16) then
              go to 12
            else
c
ccc           Calculate the row multiplier:
c
              rwmplier = XprimeX(j,i) / XprimeX(i,i)
c
ccc           Do a row operation on row 'j' using row 'i':
c
              do 11 k = 1,columns
                XprimeX(j,k) = XprimeX(j,k) - (rwmplier * XprimeX(i,k))
                XprXInvr(j,k)=XprXInvr(j,k)-(rwmplier*XprXInvr(i,k))
  11          continue
            endif
          endif
  12     continue
  13  continue
c
c     Finish the job by dividing the matrices by the diagonal elements:
c
      do 15 i = 1,totlcols
        if (XprimeX(i,i) .eq. 0) then
          go to 15
        endif
        do 14 j = 1,totlcols
          XprXInvr(i,j) = XprXInvr(i,j) / XprimeX(i,i)
  14    continue
        XprimeX(i,i) = XprimeX(i,i) / XprimeX(i,i)
  15  continue
      return
c
c     Another one done!?!!
```

```
c
      end
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
      Subroutine CmpStats(totlrows,totlcols,objctive,choice)
c                                                                       c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:
c
c
c       Variables:
c
c
      double precision Bprime(5,153)
      common /Bprime/Bprime
      double precision YprimeY,BprXprY,XprimeY(153,5),YprllprY
      common /statmtcs/YprimeY,BprXprY,XprimeY,YprllprY
      double precision SSE(5),SSR(5),SSTO(5),Rsqrd(5),AdjRsqrd(5),
     &     MSR(5),MSE(5),numrtrdf(5),dnmrtrdf(5),totaldfs(5),Fvalue(5),
     &     ParmSSQs(153,5)
      common /statstcs/SSE,SSR,SSTO,Rsqrd,AdjRsqrd,MSR,MSE,numrtrdf,
     &                 dnmrtrdf,totaldfs,fvalue,ParmSSQs
      double precision Y(385,5)
      common /rspnsmtc/Y
      double precision XtimesB(385,5)
      common /XtimesB/XtimesB
      double precision Residual(385,5)
      common /residual/Residual
      integer*2 totldlts
      common /totldlts/totldlts
      integer*2 totlrows,totlcols,objctive,choice
      integer*2 i,j,k
c
c       Begin:
c
      j = objctive
      go to (1,2),choice
c
c     Compute the parameters' sum of squares:
c
    1 do 18 i = 1,totlcols
        ParmSSQs(i,j) = Bprime(j,i) * XprimeY(i,j)
   18 continue
      return
c
c       Compute the SSE, SSR, and SSTO:
c
    2 SSE(j) = YprimeY - BprXprY
```

```
      SSR(j) = BprXprY - (YprllprY / totlrows)
      SSTO(j) = YprimeY - (YprllprY / totlrows)
c
c     Compute the R-squared and Adjusted R-squared statistics:
c
      Rsqrd(j) = 1.0 - (SSE(j) / SSTO(j))
c      AdjRsqrd(j) = 1.0 - (((totlrows - 1) * SSE(j)) /
c     &               ((totlrows - totlcols + totldlts) * SSTO(j)))
c
c     Compute the degrees of freedom for the SSTO, SSR, and SSE respectively:
c
c      totaldfs(j) = totlrows - 1
c      numrtrdf(j) = totlcols - 1.0 - totldlts
c      dnmrtrdf(j) = totaldfs(j) - numrtrdf(j)
c
c     Compute the MSR and MSE statistics:
c
c      MSR(j) = (SSR(j) / numrtrdf(j))
c      MSE(j) = (SSE(j) / dnmrtrdf(j))
c
c     Compute the F-value for the betas' test:
c
c      Fvalue(j) = MSR(j) / MSE(j)
c
c       Compute the residuals:
c
      do 11 i = 1,totlrows
        Residual(i,j) = Y(i,j) - XtimesB(i,j)
   11 continue
c
c     Sort the parameters by their sums of squares and then calculate their
c     percentage contribution to the SSR:
c
      call SortSSQs(ParmSSQs,totlcols,objctive)
      call LeastVrs(SSR,totlcols,objctive)
      return
c
c     I'm out of here!!
c
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine SortSSQs(ParmSSQs,totlcols,objctive)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  To sort the parameters' sum of squares by highest quantity.
c                 These rankings are used to delete uncontributing variables.
c
```

```
c
c        Variables:
c
c
      double precision ParmSSQs(153,5)
      double precision SrtdSSQs(153,5)
      common /SrtdSSQs/SrtdSSQs
      character varname1(153)*8,varname2(153)*8,times(153)*3
      common /varnames/varname1,varname2,times
      character srtname1(153,5)*8,srttimes(153,5)*3,srtname2(153,5)*8
      common /SrtdVars/srtname1,srttimes,srtname2
      integer*2 i,j,totlcols,objctive
      logical anyexchg
c
c     Begin:
c
      j = objctive
      do 2 i = 1,totlcols
        SrtdSSQs(i,j) = ParmSSQs(i,j)
        srtname1(i,j) = varname1(i)
        srttimes(i,j) = times(i)
        srtname2(i,j) = varname2(i)
    2 continue
      anyexchg = .true.
    3 continue
      if (anyexchg) then
        anyexchg = .false.
          do 4 i = 2,(totlcols-1)
            if (SrtdSSQs(i,j) .gt. SrtdSSQs(i+1,j)) then
              call SwapNmbr(SrtdSSQs(i,j),SrtdSSQs(i+1,j))
              call SwpChtcr(srtname1(i,j),srttimes(i,j),srtname2(i,j),
     &                 srtname1(i+1,j),srttimes(i+1,j),srtname2(i+1,j))
              anyexchg = .true.
            endif
    4     continue
        go to 3
      endif
c
c     All sorting done.  Have a sorting-good day:
c
      return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine SwapNmbr(arg1,arg2)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c        Purpose:
```

```
c
c
c         Variables:
c
c
      double precision arg1,arg2,temp
c
c     Begin:
c
      temp = arg1
      arg1 = arg2
      arg2 = temp
c
c     A quick hello -- goodbye:
c
      return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine SwpChtcr(arg11,arg12,arg13,arg21,arg22,arg23)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c         Purpose:
c
c
c         Variables:
c
c
      character*8 arg11,arg21,arg13,arg23,temp1,temp3
      character*3 arg12,arg22,temp2
c
c     Begin:
c
      temp1 = arg11
      arg11 = arg21
      arg21 = temp1
      temp2 = arg12
      arg12 = arg22
      arg22 = temp2
      temp3 = arg13
      arg13 = arg23
      arg23 = temp3
c
c     A lesser-quick hello -- goodbye:
c
      return
      end
c
```

```
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                               c
      Subroutine LeastVrs(SSR,totlcols,objctive)
c                                                                               c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  To take the sorted parameters and group them in bunches
c                 according to their percentage with respect to their SSR.
c
c
c       Variables:
c
c
      double precision SrtdSSQs(153,5)
      common /SrtdSSQs/SrtdSSQs
      double precision prcntage(153,5),ttlprcnt(153,5)
      common /prcntage/prcntage,ttlprcnt
      double precision SSR(5)
      integer*2 totlcols,objctive
      integer*2 i,j,k
c
c     Begin:
c
      j = objctive
      ttlprcnt(1,j) = 0.0
      k = 1
      do 1 i = 2,totlcols
        prcntage(i,j) = SrtdSSQs(i,j) / SSR(j)
        ttlprcnt(i,j) = ttlprcnt(i-1,j) + prcntage(i,j)
    1 continue
c
c     See you later, alligator; after while, crocidile ---
c
      return
      end
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                               c
      Subroutine RgrsOput(totlcols,totlrows,process,response)
c                                                                               c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:
c
c
c       Variables:
c
```

```
c
      integer*2 totlcols,totlrows,process,response(5)
      integer*2 i,begin,end,lookat,largest
      integer HowMany
      character*1 output,prntalso
c
c     Begin:
c
c     Allow the user to choose which objective to output.
c
      call WhichWay(output)
      lookat = HowMany(process)
      if (lookat .eq. process) then
        begin = 1
        end = process
      else
        largest = response(1)
        do 1 i = 2,process
          if (response(i) .gt. largest) then
            largest = response(i)
          endif
    1   continue
        begin = 1
        end = lookat
      endif
      if (output .eq. 'p') then
        call ToPrnter(totlcols,totlrows,process,response,largest,begin,
     &                end)
      else
        call ToScreen(totlcols,totlrows,process,response,largest,begin,
     &                end,prntalso)
        if (prntalso .eq. 'y' .or. prntalso .eq. 'Y') then
          lookat = HowMany(process)
          if (lookat .eq. process) then
            begin = 1
            end = process
          else
            largest = response(1)
            do 2 i = 2,process
              if (response(i) .gt. largest) then
                largest = response(i)
              endif
    2       continue
            begin = 1
            end = lookat
          endif
          call ToPrnter(totlcols,totlrows,process,response,largest,
     &                  begin,end)
        endif
      endif
c
c     No more output.  Bye.
c
```

```
      return
      end
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                             c
      Integer Function HowMany(process)
c                                                                             c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  Find out how many objectives the user wishes to print.
c
c
c       Variables:
c
c
      integer*2 process
      character*1 readcr
c     Begin:
c
    1 call ScrlScrn(11)
      print 3, process
      call ScrlScrn(11)
      read (*,4,err=2) HowMany
      if (HowMany .lt. 1 .or. HowMany .gt. process) then
    2   call ScrlScrn(11)
        print 5, process
        call ScrlScrn(11)
        accept 6, readcr
        go to 1
      endif
c
c     Format statements:
c
    3 format (' ','How many of the ',I1,' processed response ',
     &'objectives do you wish to output?')
    4 format (I1)
    5 format (' ','Input has to be an integer between 1 and ',I1,'!!',
     &'  Press "return" to reenter:')
    6 format (a)
c
c     Another one bites the dust!!
c
      return
      end
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                             c
      Subroutine WhichWay(output)
```

```
c                                                                         c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c      Purpose:
c
c
c      Variables:
c
c
      character*1 readcr,output
c
c     Begin:
c
    1 call ScrlScrn(10)
      print 2
      call ScrlScrn(10)
      accept 3, output
      if (output .eq. 'p' .or. output .eq. 'P') then
        output = 'p'
      else
        if (output .eq. 's' .or. output .eq. 'S') then
          output = 's'
        else
          call ScrlScrn(11)
          print 4
          call ScrlScrn(11)
          accept 3, readcr
          go to 1
        endif
      endif
c
c     Format statements:
c
    2 format (' ','Enter the letter "p" if you wish to print the',
     &' regression output. (Actually,',/,' the output is written',
     &' to a file for you to print later.)  Enter the letter',/,
     &' "s" if you wish to immediately view the regression output',
     &' via this scope.')
    3 format (a)
    4 format (' Wronge character!!!  Press "return" to try again.')
c
c     Through with this one:
c
      return
      end
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                         c
      Subroutine ToPrnter(totlcols,totlrows,process,response,largest,
     &                    begin,end)
```

```
c                                                                       c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:
c
c
c       Variables:
c
c
      double precision prcntage(153,5),ttlprcnt(153,5)
      common /prcntage/prcntage,ttlprcnt
      double precision SrtdSSQs(153,5)
      common /SrtdSSQs/SrtdSSQs
      character srtname1(153,5)*8,srttimes(153,5)*3,srtname2(153,5)*8
      common /SrtdVars/srtname1,srttimes,srtname2
      double precision SSE(5),SSR(5),SSTO(5),Rsqrd(5),AdjRsqrd(5),
     &     MSR(5),MSE(5),numrtrdf(5),dnmrtrdf(5),totaldfs(5),Fvalue(5),
     &     ParmSSQs(153,5)
      common /statstcs/SSE,SSR,SSTO,Rsqrd,AdjRsqrd,MSR,MSE,numrtrdf,
     &                 dnmrtrdf,totaldfs,fvalue,ParmSSQs
      character varname1(153)*8,varname2(153)*8,times(153)*3
      common /varnames/varname1,varname2,times
      double precision B(153,5)
      common /theBmtrx/B
      integer*2 rspnstkn(5)
      common /taken/rspnstkn
      character*2 dpndntnm(5)
      common /dpndntnm/dpndntnm
      double precision Y(385,5)
      common /rspnsmtc/Y
      double precision XtimesB(385,5)
      common /XtimesB/XtimesB
      double precision Residual(385,5)
      common /residual/Residual
      integer*2 totlcols,totlrows,process,response(5),largest,begin,end
      integer*2 i,j,k,leftover,prntdlns,sixty
      data rspnstkn/5*0/
      data sixty/60/
c
c     Begin:
c
      open(unit=7,file='regress.dat',status='old')
      do 8 i = begin,end
        if (process .eq. end) then
          j = response(i)
        else
          call WhchRspn(process,response,largest,j,i)
        endif
        write (7,900) dpndntnm(j)
        write (7,901)
        write (7,902)
        write (7,903) SSR(j),SSE(j),SSTO(j),Rsqrd(j)
```

```
        write (7,912) dpndntnm(j)
        write (7,913)
        write (7,914)
        prntdlns = 20
        do 3 k = 2,totlcols
          if (prntdlns .lt. sixty) then
            prntdlns = prntdlns + 1
            write (7,916) k-1,srtname1(k,j),srttimes(k,j),srtname2(k,j),
     &                     SrtdSSQs(k,j),prcntage(k,j),ttlprcnt(k,j)
          else
            write (7,912) dpndntnm(j)
            write (7,913)
            write (7,914)
            write (7,916) k-1,srtname1(k,j),srttimes(k,j),srtname2(k,j),
     &                     SrtdSSQs(k,j),prcntage(k,j),ttlprcnt(k,j)
            prntdlns = 11
          endif
   3    continue
        leftover = sixty - prntdlns
        if (leftover .lt. 10) then
          do 2 k = 1,leftover
            write (7,911)
   2      continue
          prntdlns = 0
        endif
        write (7,904) dpndntnm(j)
        write (7,905)
        prntdlns = prntdlns + 8
        do 1 k = 1,totlrows
          if (prntdlns .lt. sixty) then
            prntdlns = prntdlns + 1
            write (7,909)  k,Y(k,j),XtimesB(k,j),Residual(k,j)
          else
            write (7,904) dpndntnm(j)
            write (7,905)
            write (7,909)  k,Y(k,j),XtimesB(k,j),Residual(k,j)
            prntdlns = 9
          endif
   1    continue
        if (i .ne. end) then
          leftover = sixty - prntdlns
          do 7 k = 1,leftover
            write (7,911)
   7      continue
        endif
   8  continue
 900  format (' ',///,T18,'Regression Output for Dependent'
     &         ' Variable: ',A2)
 901  format (' ',//,T8,' Model Sum of',T27,' Error Sum of',T46,
     &         ' Total Sum of')
 902  format (' ',T8,'Squares (SSR)',T27,'Squares (SSE)',T46,
     &         'Squares (SSTO)',T66,'R-Square')
 903  format (' ',/,1X,3(9X,F8.2,2X),T67,F6.4)
```

```
 904  format (' ',///,T15,'For ',A2,':  Residuals Associated With The',
     &         ' Design Points')
 905  format (' ',//,T4,'Design Point',T20,'ABM''s Response',T38,
     &         'Postulated Model''s Response',T69,'Residual',/)
 909  format (' ',T8,I3,T24,F6.2,T49,F6.2,T70,F6.2)
 911  format (' ')
 912  format (' ',///,T20,'Parameters Of ',A2,' Sorted In Ascending',
     &' Order',/,T30,'By Their Sum Of Squares')
 913  format (' ',//,T39,'Type I Sum',T54,'Percent',T65,
     &'Summed Percent')
 914  format (' ',T5,'Number',T15,'Variable',T39,'of Squares',
     &T55,'Of SSR',T69,'Of SSR',/)
 916  format (' ',T5,I3,T15,A8,A3,A8,T38,F11.3,T56,2PF5.2,T69,2PF6.2)
      close (7)
c
c     No more to print.  Bye.
c
      return
      end
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
      Subroutine WhchRspn(process,response,largest,j,i)
c                                                                       c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:
c
c
c       Variables:
c
c
      integer*2 rspnstkn(5)
      common /taken/rspnstkn
      integer*2 process,response(5),largest,j,i
      integer*2 k
      character ending(5)*2,readcr*1
      logical valid
      data ending/'st','nd','rd',2*'th'/
c
c     Begin:
c
  1   call ScrlScrn(11)
      print 5, i,ending(i)
      call ScrlScrn(11)
      read (*,6,err=2) j
      if (j .lt. 1 .or. j .gt. largest) then
c
c       If user's input is not an integer and not within the bounds,
c       make him/her reenter.
```

```
c
   2    call ScrlScrn(11)
        print 7, largest
        call ScrlScrn(11)
        accept 8, readcr
        go to 1
      endif
      rspnstkn(i) = j
      valid = .false.
c      do 55 k = 1,5
c        print *, response(k)
c  55  continue
c      print *, process,j
      do 3 k = 1,process
        if (j .eq. response(k)) then
          valid = .true.
        endif
   3  continue
      if (valid) then
        do 4 k = 1,i
          if (k .ne. i) then
            if (rspnstkn(k) .eq. rspnstkn(i)) then
c
c             If the user tries to process a response twice, send him/her
c             back to the beginning to reenter his/her answer.
c
              call ScrlScrn(10)
              print 9, j,k,ending(k)
              call ScrlScrn(11)
              accept 8, readcr
              go to 1
            endif
          endif
   4    continue
      else
        call ScrlScrn(11)
        print 10, j,ending(j)
        call ScrlScrn(10)
        accept 8, readcr
        go to 1
      endif
c
c     Format Statements:
c
   5  format (' ','What is the ',I1,A2,' response objective you want',
     &' to output?')
   6  format (I)
   7  format (' ','Value has to be an integer between 1 and ',I1,'!!',
     &       /,' Press "return" to reenter value:')
   8  format (a)
   9  format (' ','You have already entered the value ',I1,' for the ',
     &I1,A2,' response objective!!',/,' Press "return" to reenter:')
  10  format (' ','Sorry!!  You did not ask for the ',I1,A2,' response',
```

```
     &' to be processed!',/,' Press "return" to reenter:')
c
c     Just checking on you:
c
      return
      end
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                               c
      Subroutine ToScreen(totlcols,totlrows,process,response,largest,
     &                    begin,end,prntalso)
c                                                                               c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:
c
c
c       Variables:
c
c
      double precision prcntage(153,5),ttlprcnt(153,5)
      common /prcntage/prcntage,ttlprcnt
      double precision SrtdSSQs(153,5)
      common /SrtdSSQs/SrtdSSQs
      character srtname1(153,5)*8,srttimes(153,5)*3,srtname2(153,5)*8
      common /SrtdVars/srtname1,srttimes,srtname2
      double precision SSE(5),SSR(5),SSTO(5),Rsqrd(5),AdjRsqrd(5),
     &     MSR(5),MSE(5),numrtrdf(5),dnmrtrdf(5),totaldfs(5),Fvalue(5),
     &     ParmSSQs(153,5)
      common /statstcs/SSE,SSR,SSTO,Rsqrd,AdjRsqrd,MSR,MSE,numrtrdf,
     &                 dnmrtrdf,totaldfs,fvalue,ParmSSQs
      character varname1(153)*8,varname2(153)*8,times(153)*3
      common /varnames/varname1,varname2,times
      double precision B(153,5)
      common /theBmtrx/B
      integer*2 rspnstkn(5)
      common /taken/rspnstkn
      character*2 dpndntnm(5)
      common /dpndntnm/dpndntnm
      double precision Y(385,5)
      common /rspnsmtc/Y
      double precision XtimesB(385,5)
      common /XtimesB/XtimesB
      double precision Residual(385,5)
      common /residual/Residual
      integer*2 totlcols,totlrows,process,response(5),largest,begin,end
      character*1 prntalso
      integer*2 i,j,k,twenty,lnsscrld,lnstoscl,nineteen
      character*1 readcr,repeat,goodchar
      data rspnstkn/5*0/
```

```
      data nineteen/19/,twenty/20/
c
c     Begin:
c
c     Allow the user to choose which objective to output.
c
      repeat = 'y'
   1  if (repeat .eq. 'y' .or. repeat .eq. 'Y') then
        do 8 i = 1,end
          rspnstkn(i) = 0
   8    continue
        do 5 i = begin,end
          if (process .eq. end) then
            j = response(i)
          else
            call WhchRspn(process,response,largest,j,i)
          endif
          call ScrlScrn(7)
          write (*,900) dpndntnm(j)
          write (*,901)
          write (*,902)
          write (*,903) SSR(j),SSE(j),SSTO(j),Rsqrd(j)
          call ScrlScrn(6)
          lnsscrld = 20
          do 3 k = 2,totlcols
            if (lnsscrld .le. nineteen) then
              lnsscrld = lnsscrld + 1
              write (*,917) k-1,srtname1(k,j),srttimes(k,j),
     &                      srtname2(k,j),SrtdSSQs(k,j),
     &                      prcntage(k,j),ttlprcnt(k,j)
            else
              call ScrlScrn(1)
              write (*,912)
              call ScrlScrn(1)
              accept 913, readcr
              write (*,914) dpndntnm(j)
              write (*,915)
              write (*,916)
              write (*,917) k-1,srtname1(k,j),srttimes(k,j),
     &                      srtname2(k,j),SrtdSSQs(k,j),
     &                      prcntage(k,j),ttlprcnt(k,j)
              lnsscrld = 8
            endif
   3      continue
          lnstoscl = nineteen - lnsscrld
          call ScrlScrn(lnstoscl)
          lnsscrld = 20
          do 2 k = 1,totlrows
            if (lnsscrld .le. nineteen) then
              lnsscrld = lnsscrld + 1
              write (*,909)  k,Y(k,j),XtimesB(k,j),Residual(k,j)
            else
              call ScrlScrn(1)
```

```
              write (*,912)
              call ScrlScrn(1)
              accept 913, readcr
              write (*,904) dpndntnm(j)
              write (*,905)
              write (*,909)  k,Y(k,j),XtimesB(k,j),Residual(k,j)
              lnsscrld = 6
            endif
 2        continue
          lnstoscl = twenty - lnsscrld
          call ScrlScrn(lnstoscl)
          write (*,912)
          call ScrlScrn(1)
          accept 913, readcr
 5      continue
 6      call ScrlScrn(11)
        print *, 'Do you wish to view your output again?  (y/n)'
        call ScrlScrn(11)
        accept 913, repeat
        call ChckChar(repeat,goodchar)
        if (goodchar .ne. 'y') then
          call BadChar
          go to 6
        endif
        go to 1
      endif
 7    call ScrlScrn(11)
      print *, 'Do you also wish to print your output?  (y/n)'
      call ScrlScrn(11)
      accept 913, prntalso
      call ChckChar(prntalso,goodchar)
      if (goodchar .ne. 'y') then
        call BadChar
        go to 7
      endif

900   format (' ',T18,'Regression Output for Dependent'
     &         ' Variable: ',A2)
901   format (' ',//,T8,' Model Sum of',T27,' Error Sum of',T46,
     &         ' Total Sum of')
902   format (' ',T8,'Squares (SSR)',T27,'Squares (SSE)',T46,
     &         'Squares (SSTO)',T66,'R-Square')
903   format (' ',/,1X,3(9X,F8.2,2X),T67,F6.4)
904   format (' ',T15,'For ',A2,':  Residuals Associated With The',
     &         ' Design Points')
905   format (' ',//,T4,'Design Point',T20,'ABM''s Response',T38,
     &         'Postulated Model''s Response',T69,'Residual',/)
909   format (' ',T8,I3,T24,F6.2,T49,F6.2,T70,F6.2)
911   format (' ')
912   format (' ','Press "return" to continue.')
913   format (a)
914   format (' ',T20,'Parameters Of ',A2,' Sorted In Ascending',
     &' Order',/,T30,'By Their Sum Of Squares')
```

```
 915  format (' ',//,T39,'Type I Sum',T54,'Percent',T65,
     &'Summed Percent')
 916  format (' ',T5,'Number',T15,'Variable',T39,'of Squares',
     &T55,'Of SSR',T69,'Of SSR',/)
 917  format (' ',T5,I3,T15,A8,A3,A8,T38,F11.3,T56,2PF5.2,T69,2PF6.2)
c
c     No more to print.  Bye.
c
      return
      end
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
      Subroutine ForDcode(paramtrs,totlcols,process,response)
c                                                                       c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  To write the parameter estimates out to a data set for input
c                 to the Decode subroutine.
c
c
c       Variables:
c
c
      character varname1(153)*8,varname2(153)*8,times(153)*3
      common /varnames/varname1,varname2,times
      double precision B(153,5)
      common /theBmtrx/B
      integer*2 response(5)
      integer*2 i,j,k,paramtrs,totlcols,process
c
c     Begin:
c
      open(unit=7,file='codedbts.in',status='old')
      write (7,11) paramtrs,totlcols,process
      do 1 k = 1,totlcols
        write (7,14) varname1(k),times(k),varname2(k)
    1 continue
      do 3 i = 1,process
        j = response(i)
        write (7,12) j
        do 2 k = 1,totlcols
          write (7,13) B(k,j)
    2   continue
    3 continue
      close (7)
c
c     Format statements:
c
   11 format (1X,I2,1X,I3,1X,I1)
```

```
   12  format (1X,I1)
   13  format (1X,F9.5)
   14  format (1X,A8,A3,A8)
c
c      I'm finally finished with the entire Rgrssion Subroutine:
c
       return
       end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                      c
       Subroutine Filerror(filename,fileindx,whichone)
c                                                                      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c        Purpose:
c
c
c        Variables:
c
c
       integer*2 whichone
       character filename*8,fileindx*4,readcr*1
c
c      Begin:
c
       call ScrlScrn(10)
       if (whichone .eq. 1) then
         print 1, filename,fileindx
       else
         print 2, filename,fileindx
       endif
       call ScrlScrn(10)
       accept 3, readcr
    1  format (' ','An error has occurred while attempting to open ',
      &'file "',A8,A4,'".',/,' Exit this program and ensure the file ',
      &'exists and is capable of being read.',/,' Press "return":')
    2  format (' ','An error has occurred while reading the data in ',
      &'file "',A8,A4,'".',/,' Exit this program and ensure that data ',
      &'exists and is in the correct format.',/,' Press "return":')
    3  format (a)
c
c      Bye, Bye:
c
       return
       end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
c                                                                          c
      Subroutine Decode
c                                                                          c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c     Purpose:  Subroutine Decode decodes the estimated parameter estimates
c               receives the regression equation's coefficient
c               estimates, which were estimated with coded input, and transforms
c               them
c
c
c     Variables:
c
c
      integer*2 i,k,paramtrs,totlcols,process,response(5),objctive,
     &          thsedone(5),prtrcntr
      character dpndntnm(5)*2,doit*1,goodchar*1
      logical filesok
      data dpndntnm/'Y1','Y2','Y3','Y4','Y5'/
c
c     begin:
c
      filesok = .true.
      call DecodeIn(paramtrs,totlcols,process,response,filesok)
      if (filesok) then
        call StghtFwd(paramtrs,totlcols,process,response)
        k = 0
        do 2 i = 1,process
          objctive = response(i)
   1      call ScrlScrn(11)
          print 3, dpndntnm(objctive)
          call ScrlScrn(11)
          accept 4, doit
          call ChckChar(doit,goodchar)
          if (goodchar .ne. 'y') then
            call BadChar
            go to 1
          endif
          if (doit .eq. 'y' .or. doit .eq. 'Y') then
            k = k + 1
            thsedone(k) = objctive
            call InitVars(paramtrs,objctive,totlcols)
            call Calclate(paramtrs,objctive)
          endif
   2    continue
        prtrcntr = k
        call DcdePrnt(paramtrs,totlcols,thsedone,dpndntnm,prtrcntr)
      endif
c
c     Format statements:
c
   3  format (' ','Do you wish to decode the parameter estimates for',
```

```
     &' response objective ',A2,'?  (y/n)')
   4  format (a)
c
c     Decode no more
c
      return
      end
c
c
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                             c
      Subroutine DecodeIn(paramtrs,totlcols,process,response,filesok)
c                                                                             c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  To obtain all input used by the Decode Subroutine.
c
c
c       Variables:
c
c
      double precision div(16),mid(16)
      common /divmids/div,mid
      double precision low(16),high(16)
      common /lowhigh/low,high
      character varname1(153)*8,varname2(153)*8,times(153)*3
      common /varnames/varname1,varname2,times
      double precision B(153,5)
      common /theBmtrx/B
      character*8 thenames(16)
      common /TheNames/thenames
      integer*2 paramtrs,totlcols,process,response(5)
      integer*2 i,j,k,whichone
      character filename*8,fileindx*4
      logical filesok
c
c     Begin:
c
      whichone = 1
      open(unit=2,err=5,file='codedbts.in',status='old')
      whichone = 2
      read (2,11,err=5) paramtrs,totlcols,process
      do 1 i = 1,totlcols
        read (2,14,err=5) varname1(i),times(i),varname2(i)
   1  continue
      do 3 i = 1,process
        read (2,12,err=5) response(i)
        j = response(i)
        do 2 k = 1,totlcols
          read (2,13,err=5) B(k,j)
   2    continue
```

```
    3  continue
       close (2)
       whichone = 1
       open (unit=3,err=6,name='decode.in',status='old')
       whichone = 2
       do 4 i = 1,paramtrs
         read (3,16,err=6) thenames(i),low(i),high(i),mid(i),div(i)
    4  continue
       close (3)
       go to 17
c
c      Error code:
c
    5  if (whichone .eq. 2) then
         close (2)
       endif
       filename = 'CODEDBTS'
       fileindx = '.IN '
       go to 7
    6  if (whichone .eq. 2) then
         close (3)
       endif
       filename = 'DECODE'
       fileindx = '.IN '
    7  call Filerror(filename,fileindx,whichone)
       filesok = .false.
c
c      Format statements:
c
   11  format (1X,I2,1X,I3,1X,I1)
   12  format (1X,I1)
   13  format (1X,F9.5)
   14  format (1X,A8,A3,A8)
   16  format (1X,A8,2(1X,F7.2),2(1X,F11.6))
c
c      I've got my input, so there:
c
   17  return
       end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
       Subroutine StghtFwd(paramtrs,totlcols,process,response)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c        Purpose:  Put the parameter estimates and names in forward position,
c                  i.e., X0 X1 X2 X3 . . . X11 X12 X13 . . . X22 X23 . . . X33.
c
c
```

```
c       Variables:
c
c
      double precision B(153,5)
      common /theBmtrx/B
      double precision frwdbeta(153,5)
      common /forwrdBs/frwdbeta
      character varname1(153)*8,varname2(153)*8,times(153)*3
      common /varnames/varname1,varname2,times
      character fwdname1(153)*8,fwdname2(153)*8,fwdtimes(153)*3
      common /fwdnames/fwdname1,fwdname2,fwdtimes
      integer*2 paramtrs,totlcols,process,response(5)
      integer*2 i,j,k,l,m,n,parmplus,indx1,indx2,limit
c
c     Begin:
c
c     Obtain the "coded" parameter estimates in forward position:
c
      parmplus = paramtrs + 1
      do 4 i = 1,process
        j = response(i)
        frwdbeta(1,j) = B(1,j)
        do 1 k = 2,parmplus
          frwdbeta(k,j) = B(paramtrs+k,j)
    1   continue
        l = 1
        indx1 = paramtrs + 1
        indx2 = (2 * paramtrs) + 1
        limit = paramtrs
        do 3 m = 1,paramtrs
          indx1 = indx1 + 1
          l = l + 1
          frwdbeta(indx1,j) = B(l,j)
          limit = limit - 1
          do 2 n = 1,limit
            indx1 = indx1 + 1
            indx2 = indx2 + 1
            frwdbeta(indx1,j) = B(indx2,j)
    2     continue
    3   continue
    4 continue
c
c     Obtain the parameter names in forward position
c
      fwdname1(1) = varname1(1)
      fwdtimes(1) = times(1)
      fwdname2(1) = varname2(1)
      do 5 k = 2,parmplus
        fwdname1(k) = varname1(paramtrs+k)
        fwdtimes(k) = times(paramtrs+k)
        fwdname2(k) = varname2(paramtrs+k)
    5 continue
      l = 1
```

```
      indx1 = paramtrs + 1
      indx2 = (2 * paramtrs) + 1
      limit = paramtrs
      do 7 m = 1,paramtrs
        indx1 = indx1 + 1
        l = l + 1
        fwdname1(indx1) = varname1(l)
        fwdtimes(indx1) = times(l)
        fwdname2(indx1) = varname2(l)
        limit = limit - 1
        do 6 n = 1,limit
          indx1 = indx1 + 1
          indx2 = indx2 + 1
          fwdname1(indx1) = varname1(indx2)
          fwdtimes(indx1) = times(indx2)
          fwdname2(indx1) = varname2(indx2)
    6   continue
    7 continue
c
c     Don't be forward anymore:
c
      return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
        Subroutine InitVars(paramtrs,objctive,totlcols)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  To init Decode's "working" variables.
c
c
c       Variables:
c
c
      double precision frwdbeta(153,5)
      common /forwrdBs/frwdbeta
      double precision cdebeta0,clinbeta(16),ccrsbeta(136)
      common /cdedtrms/cdebeta0,clinbeta,ccrsbeta
      integer*2 paramtrs,objctive,totlcols
      integer*2 i,j,indx,crsparms
c
c     Begin:
c

      j = objctive
c
c     Obtain the "working" Beta0 first:
c
```

```
      cdebeta0 = frwdbeta(1,j)
c
c     Obtain the "working" lineara betas second:
c
      do 1 i = 1,paramtrs
        clinbeta(i) = frwdbeta(i+1,j)
    1 continue
c
c     Compute the number of pure and cross terms; then obtain the "working"
c     quadratic betas third:
c
      crsparms = totlcols - (paramtrs + 1)
      indx = paramtrs + 1
      do 2 i = 1,crsparms
        ccrsbeta(i) = frwdbeta(indx+i,j)
    2 continue
c
c     My "working" days are over:
c
      return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                          c
      Subroutine Calclate(paramtrs,objctive)
c                                                                          c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  To calculate the decoded parameter estimates.
c
c
c       Variables:
c
c
      double precision div(16),mid(16)
      common /divmids/div,mid
      double precision cdebeta0,clinbeta(16),ccrsbeta(136)
      common /cdedtrms/cdebeta0,clinbeta,ccrsbeta
      double precision beta0(5),beta(16,5),crssbeta(136,5)
      common /decdedBs/beta0,beta,crssbeta
      integer*2 paramtrs,objctive
      double precision clintrms,ccrstrms,tccrstrm
      integer*2 i,j,k,inc1,inc2,inc3,adjust1,adjust2,indx
c
c     Begin:
c
c     Calculate beta0
c
      k = objctive
      clintrms = 0.0
```

```
      tccrstrm = 0.0
      indx = 0
      do 12 i = 1,paramtrs
        ccrstrms = 0.0
        clintrms = clintrms + (clinbeta(i) * (mid(i) / div(i)))
        do 13 j = i,paramtrs
          indx = indx + 1
          ccrstrms = ccrstrms + (ccrsbeta(indx) * mid(j) / div(j))
   13   continue
        ccrstrms = ccrstrms * (mid(i) / div(i))
        tccrstrm = tccrstrm + ccrstrms
   12 continue
      beta0(k) = cdebeta0 - clintrms + tccrstrm
c
c     Calculate the linear betas
c
      clintrms = 0.0
      ccrstrms = 0.0
      inc1 = 0
      inc2 = 1
      adjust2 = paramtrs + 1
      do 14 i = 1,paramtrs
        inc1 = i
        adjust1 = paramtrs
        do 15 j = 1,paramtrs
          if (j .lt. i) then
            ccrstrms = ccrstrms + (ccrsbeta(inc1) * mid(j) / div(j))
            adjust1 = adjust1 - 1
            inc1 = inc1 + adjust1
          endif
          if (i .eq. j) then
            ccrstrms = ccrstrms + (ccrsbeta(inc2) * 2 * mid(i) / div(i))
            inc3 = inc2
            adjust2 = adjust2 - 1
            inc2 = inc2 + adjust2
          endif
          if (j .gt. i) then
            inc3 = inc3 + 1
            ccrstrms = ccrstrms + (ccrsbeta(inc3) * mid(j) / div(j))
          endif
   15   continue
        beta(i,k) = (clinbeta(i) - ccrstrms) / div(i)
        ccrstrms = 0.0
   14 continue
c
c     Calculate the squared and cross termed betas
c
      indx = 0
      do 16 i = 1,paramtrs
        do 17 j = i,paramtrs
          indx = indx + 1
          crssbeta(indx,k) = ccrsbeta(indx) / (div(i) * div(j))
   17   continue
```

```
  16  continue
c
c     Calc or not to Calc --
c
      return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine DcdePrnt(paramtrs,totlcols,thsedone,dpndntnm,prtrcntr)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:  To print the decoded parameter estimates.
c
c
c       Variables:
c
c
      character fwdname1(153)*8,fwdname2(153)*8,fwdtimes(153)*3
      common /fwdnames/fwdname1,fwdname2,fwdtimes
      double precision beta0(5),beta(16,5),crssbeta(136,5)
      common /decdedBs/beta0,beta,crssbeta
      double precision low(16),high(16)
      common /lowhigh/low,high
      character*8 thenames(16)
      common /TheNames/thenames
      integer*2 paramtrs,totlcols,thsedone(5),prtrcntr
      character*2 dpndntnm(5)
      integer*2 i,j,k,indx,crsstrms,leftover,prntdlns,sixty
      data sixty/60/
c
c     Begin:
c
c     Write out important info for the "XCUTEQNS" subroutine:
c
      open (unit=4,name='xcuteqns.in',status='old')
      write (4,7) paramtrs,totlcols,prtrcntr
      do 8 i = 1,paramtrs
        write (4,9) thenames(i),low(i),high(i)
   8  continue
c
c     Start the ball a-rolling:
c
      open (unit=7,name='decode.out',status='old')
      do 12 k =1,prtrcntr
        j = thsedone(k)
        write (4,10) j
c
c       Now, write out the decoded betas in print format (60 lines):
```

```
c
c
c       Print heading:
c
        write (7,4) dpndntnm(j)
        prntdlns = 6
c
c       Print Beta0 (the intercept):
c
        write (7,5), fwdname1(1),fwdtimes(1),fwdname2(1),beta0(j)
        write (4,11), beta0(j)
        prntdlns = prntdlns + 1
c
c       Print the linear betas:
c
        indx = 1
        do 1 i = 1,paramtrs
          indx = indx + 1
          write (7,5), fwdname1(indx),fwdtimes(indx),fwdname2(indx),
     &                 beta(i,j)
          write (4,11), beta(i,j)
          prntdlns = prntdlns + 1
   1    continue
c
c      Print the quadratic betas:
c
        crsstrms = totlcols - (paramtrs + 1)
        do 2 i = 1,crsstrms
          if (prntdlns .lt. sixty) then
            indx = indx + 1
            write (7,5), fwdname1(indx),fwdtimes(indx),fwdname2(indx),
     &                   crssbeta(i,j)
            write (4,11), crssbeta(i,j)
            prntdlns = prntdlns + 1
          else
            indx = indx + 1
            write (7,4) dpndntnm(j)
            write (7,5), fwdname1(indx),fwdtimes(indx),fwdname2(indx),
     &                   crssbeta(i,j)
            write (4,11), crssbeta(i,j)
            prntdlns = 7
          endif
   2    continue
c
c       Finish the page out:
c
        leftover = sixty - prntdlns
        do 3 i = 1,leftover
          write (7,6)
   3    continue
  12  continue
c
c     Format statements:
```

```
c
    4  format (' ',///,'The decoded parameter estimates for ',A2,' are:',
      &//)
    5  format (' ',2X,A8,A3,A8,2X,F18.12)
    6  format (' ')
    7  format (1X,I2,1X,I3,1X,I1)
    9  format (1X,A8,2(1X,F7.2))
   10  format (1X,I1)
   11  format (1X,F18.12)
       close (4)
       close (7)
c
c      I'm printing my life away --
c
       return
       end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                                c
       Subroutine Xcuteqns
c                                                                                c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c        Purpose:
c
c
c        Variables:
c
c
       character dpndntnm(5)*2,yesno*1,goodchar*1,repeat*1
       integer*2 i,paramtrs,totlcols,process,response(5),objctive
       data dpndntnm/'Y1','Y2','Y3','Y4','Y5'/
c
c      Begin:
c
       call XecuteIn(paramtrs,totlcols,process,response)
       do 4 i = 1,process
         objctive = response(i)
    1    call ScrlScrn(11)
         print 5, dpndntnm(objctive)
         call ScrlScrn(11)
         accept 6, yesno
         call ChckChar(yesno,goodchar)
         if (goodchar .ne. 'y') then
           call BadChar
           go to 1
         endif
         if (yesno .eq. 'y' .or. yesno .eq. 'Y') then
           repeat = 'y'
    2      if (repeat .eq. 'y' .or. repeat .eq. 'Y') then
```

```
            call UserInpt(paramtrs,totlcols)
            call SetUpEqn(paramtrs)
            call DoandShw(objctive,dpndntnm,totlcols)
3           call ScrlScrn(11)
            print 7, dpndntnm(objctive)
            call ScrlScrn(11)
            accept 6, repeat
            call ChckChar(repeat,goodchar)
            if (goodchar .ne. 'y') then
              call BadChar
              go to 3
            endif
            go to 2
          endif
        endif
4     continue
5     format (' ','Do you wish to execute response objective ',A2,'?',
     &'  (y/n)')
6     format (a)
7     format (' ','Do you wish to execute response objective ',A2,
     &' again?  (y/n)')
c
c     So long; till nex time:
c
      return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
      Subroutine XecuteIn(paramtrs,totlcols,process,response)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:
c
c
c       Variables:
c
c
      double precision thebetas(153,5)
      common /XecuteBs/thebetas
      double precision low(16),high(16)
      common /lowhigh/low,high
      character*8 thenames(16)
      common /TheNames/thenames
      integer*2 i,j,k,paramtrs,totlcols,process,response(5)
c
c     Begin:
c
      open (unit=4,name='xcuteqns.in',status='old')
```

```
      read (4,7) paramtrs,totlcols,process
      do 1 i = 1,paramtrs
        read (4,9) thenames(i),low(i),high(i)
    1 continue
      do 3 i = 1,process
        read (4,10) response(i)
        j = response(i)
        do 2 k = 1,totlcols
          read (4,11) thebetas(k,j)
    2   continue
    3 continue
c
c
    7 format (1X,I2,1X,I3,1X,I1)
    9 format (1X,A8,2(1X,F7.2))
   10 format (1X,I1)
   11 format (1X,F18.12)
      close (4)
      return
      end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                            c
      Subroutine UserInpt(paramtrs,totlcols)
c                                                                            c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c       Purpose:
c
c
c       Variables:
c
c
      double precision low(16),high(16)
      common /lowhigh/low,high
      character*8 thenames(16)
      common /TheNames/thenames
      double precision varvlues(16)
      common /varvlues/varvlues
      integer*2 i,paramtrs,totlcols
      character*1 readcr
c
c     Begin:
c
      do 3 i = 1,paramtrs
    1   call ScrlScrn(11)
        print 4, low(i),high(i),thenames(i)
        call ScrlScrn(11)
        read (*,*,err=2) varvlues(i)
        if (varvlues(i) .gt. high(i) .or. varvlues(i) .lt. low(i)) then
```

```
    2      call ScrlScrn(11)
           print 5, low(i),high(i)
           call ScrlScrn(10)
           accept 6, readcr
           go to 1
         endif
    3  continue
c
c      Format statements:
c
    4  format (' Within the range of ',F7.2,' and ',F7.2,', what is ',
      &'the value for "',A8,'"?')
    5  format (' Input has to be a real value within the limits of ',
      &F7.2,' and ',F7.2'.',/,' Press "return" to reenter.')
    6  format (a)
c
c      Goodbye user:
c
       return
       end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
       Subroutine SetUpEqn(paramtrs)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c        Purpose:
c
c
c        Variables:
c
c
       double precision varvlues(16)
       common /varvlues/varvlues
       double precision thevlues(2:153)
       common /thevlues/thevlues
       integer*2 i,j,indx,paramtrs
c
c      Begin:
c
       do 1 i = 1,paramtrs
         thevlues(i+1) = varvlues(i)
    1  continue
       indx = paramtrs + 1
       do 3 i = 1,paramtrs
         do 2 j = i,paramtrs
           indx = indx + 1
           thevlues(indx) = (varvlues(i) * varvlues(j))
    2    continue
```

```
    3  continue
c
c      So long:
c
       return
       end
c
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
       Subroutine DoandShw(objctive,dpndntnm,totlcols)
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c        Purpose:
c
c
c        Variables:
c
c
       double precision thebetas(153,5)
       common /XecuteBs/thebetas
       double precision thevlues(2:153)
       common /thevlues/thevlues
       double precision theanswr
       integer*2 i,j,objctive,totlcols
       character*1 readcr,dpndntnm(5)*2
c
c      Begin:
c
       j = objctive
       theanswr = thebetas(1,j)
       do 1 i = 2,totlcols
         theanswr = theanswr + (thevlues(i) * thebetas(i,j))
    1  continue
       call ScrlScrn(11)
       print 2, dpndntnm(j),theanswr
       call ScrlScrn(11)
       accept 3, readcr
c
c      Format statements:
c
    2  format (' ','For ',A2,', survey says:  ',F6.2,/,
      &' Press "return":')
    3  format (a)
c
c      Parting is such sweet sorry:
c
       return
       end
```

Appendix C: RSMed-AEM Program User's Guide

Since the automation program is written in Fortran-77 and executed on the VAX 11/785, the program is started by typing "RUN RSMEDAEM.EXE" on a CRT terminal. The automated RSM program is designed to interact with the user, so, after starting the program, a main menu will appear (Figure 4.9). The main menu is used as a guide to show the sequential execution of the RSMed-AEM program.

Input Data

By selecting I, the INPTDATA subroutine is invoked allowing the user to enter all pertinent data (Figures 4.6 and 4.8) for the variables of interest (design variables). Once the INPTDATA subroutine is executed, the user is asked a series of questions, dependent upon the user's last response.

The user is first asked whether or not the characteristics of the design variables already reside in the VARIBLES.DAT file. If the design variable information does not exist, the program queries the user for the information. The user is asked to input the number of design variables, their names, their low and high values, and the number of AEM variables each design variable controls. All input is checked for logic and limit errors. An example of a logic error is a design variable's high value can not be less than its low value. A limit error will occur when no Box-Behnken experimental design exists for the number of design variables of interest. If the user can not use the existing experimental designs, he/she is allowed to exit the INPTDATA subroutine.

Whether the characteristics of the design variables already reside in the VARIBLES.DAT file or is entered via the CRT, the information is gathered in a table and shown to the user via the CRT. The user is asked to verify that all data is correct. If not, the user is allowed to change either a variable's name, low/high value, or the number of AEM variables it controls.

Once verification is complete, the INPTDATA subroutine creates three files (CODEDSGN.IN, DECODE.IN, and RGRSSION.IN) that are used later in the RSMed-AEM program (refer to Appendices G, H, and I). After those three files are created, the user is allowed the opportunity to save the design variable information in the file VARIBLES.DAT. Any information already in that file is written over. The main menu reappears once the INPTDATA subroutine finishes its tasks.

Build Uncoded/Coded Design

By selecting either U or C from the main menu, the CODEDSGN subroutine is invoked. The CODEDSGN subroutine does not interact with the user. Its function is to create an experimental design file, coded or uncoded, dependent upon which selection, C or U respectively, the user picks.

With the selection of U, the CODEDSGN subroutine reads the information from the CODEDSGN.IN file and creates an uncoded experimental design file entitled AEM.DAT (Appendix J). Once the AEM.DAT file is created, the main menu reappears. The RSMed-AEM program should now be exited (i.e., selection E), and the DATACLCT.COM program executed (Tool Two of Appendix D).

With the selection of C, the CODEDSGN subroutine reads the information from the CODEDSGN.IN file and creates a coded experimental design file entitled DSGNMTRX.IN (Appendix N) to be used with the RGRSSION subroutine. Once the DSGNMTRX.IN file is created, the main menu reappears. **Caution**: Do not build the uncoded and coded files at the same time. Build the uncoded file first (selection U); exit the RSMed-AEM program (selection E) to use the DATACLCT.COM program (Appendix D); and then restart the RSMed-AEM program to build the coded file (selection C). To do else may result in an error.

Regression

By selecting R from the main menu, the RGRSSION subroutine is executed. The RGRSSION subroutine uses previously created files to calculate some regression information. The information consists of the model, error, and total sum-of-squares statistics, the R-Squared statistic, and the parameter estimate for each term in the postulated equation. Also, to help the user decide how to reduce a full postulated equation, the terms of the postulated equation are sorted in ascending order by their contribution to the model sum-of-squares. Finally, a listing of the AEM response, postulated model response, and their difference (residual) is listed per design point.

After invoking the RGRSSION subroutine, the RGRSSION.IN and DSGNMTRX.IN files are used for input (Appendices I and N respectively). However, the RGRSSION subroutine also requires input through user interaction. The first question asked the user is "How many response objectives (total) are there? (max of 5)". The RGRSSION subroutine needs to read the RESPNSES.IN file (Appendix M) but first needs to know

how many objectives per line there are in the file. Therefore, the user must input the total number of objectives that are in that file.

After all input files are read, the program continues to query the user for more information. The user is asked to supply how many of the total objectives, and in what order he/she wishes to process them through the RGRSSION subroutine. Also, the user is allowed the ability to delete terms, except the INTERCEPT, if he/she so desires. (This ability is used for reducing the full postulated model.) Next, the user is given the choice of viewing the regression output directly by screen and/or writing it to file for printing later (REGRESS.DAT -- Appendix O). Finally, for output, the user is allowed to view/print all or just some of the processed objectives.

After all user interaction is complete, the CODEDBTS.IN file (Appendix P) is created for use with the DECODE subroutine. The CODEDBTS.IN file contains the parameter estimates for each term in the postulated equation, only in coded form. Once the RGRSSION subroutine is complete, the main menu reappears.

Decode

By selecting D, the DECODE subroutine is started. The DECODE subroutine decodes the coded parameter estimates from the CODEDBTS.IN file. The DECODE.IN and CODEDBTS.IN files are used as input (Appendices H and P respectively). Interactive input from the user is used for controlling which objectives are decoded. The decoded output is written to two files: DECODE.OUT for possible printing and XCUTEQNS.IN for use with the XCUTEQNS subroutine (Appendix T and S respectively). Once the DECODE subroutine is finished, control is passed back to the main menu.

Execute Equations

A selection of X from the main menu executes the XCUTEQNS subroutine. The XCUTEQNS subroutine is not a part of the automated RSMed-AEM process even though it is included within the RSMed-AEM program. The XCUTEQNS subroutine is an added feature that gives the user the ability to execute the estimated equation. The XCUTEQNS subroutine reads the XCUTEQNS.IN (Appendix S) file as input and sets up a matrix containing the terms to each response objective's estimated equation (as long as that response objective was processed through the RGRSSION subroutine). The user is next asked to interactively add to the input. He/she is first asked which response objective's estimated equation is to be executed. Next, the user is asked to input a value for each design variable. The RSMed-AEM program insures the user input is within the bounds for each design variable. The output, the response to the estimated equation given the variable values, is printed to the screen. The user can continue executing the estimated equation until he/she is finished. Once finished, the main menu reappears.

Appendix D: Tool Two -- The DATACLCT.COM Program

The DATACLCT.COM program is written in the VAX 11/785 digital command language. DATACLCT.COM controls the data collection, used with the RGRSSION subroutine of the RSMed-AEM program, by automatically executing the AEM and then collecting and storing the required data from each execution in a file called STORAGE.DAT (Appendix L). The DATACLCT.COM program uses the IAEM1.DAT file (Appendix E) and the AEM.DAT file (Appendix J) as inputs.

```
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT ""
$ INQUIRE REPS "Enter the number of repetitions and press return"
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT "Do you wish to be queried on program continuation after "
$ INQUIRE QUERY "each repetition?  (y/n)"
$ FIRST = "  TNUMB(1,1)"
$ SECOND = "  TNUMB(15,1"
$ THIRD = "+          "
$ COUNTER1 = 0
$LOOP1:
$ COUNTER1 = COUNTER1 + 1
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT "Repetition Number ''COUNTER1' "
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT ""
$ OPEN UPDTDATA AEM.IN
$ OPEN/WRITE TMPFILE1 TEMP1.DAT
$ READ UPDTDATA REC1
$ READ UPDTDATA REC2
$LOOP2:
$ READ/END_OF_FILE=ENDLOOP2 UPDTDATA REC3
$ WRITE TMPFILE1 REC3
$ GOTO LOOP2
$ENDLOOP2:
$ CLOSE UPDTDATA
$ CLOSE TMPFILE1
$ COPY TEMP1.DAT AEM.IN
$ OPEN AEMINPUT IAEM1.DAT
$ OPEN/WRITE TMPFILE2 TEMP2.DAT
```

```
$ COUNTER2 = 0
$LOOP3:
$ READ/END_OF_FILE=ENDLOOP3 AEMINPUT TRNSFREC
$ COUNTER2 = COUNTER2 + 1
$ IF COUNTER2 .EQ. 41 .OR. COUNTER2 .EQ. 42 THEN GOTO INSRTREC
$ WRITE TMPFILE2 TRNSFREC
$ GOTO LOOP3
$ INSRTREC:
$ IF COUNTER2 .EQ. 41 THEN WRITE TMPFILE2 REC1
$ IF COUNTER2 .EQ. 42 THEN WRITE TMPFILE2 REC2
$ GOTO LOOP3
$ENDLOOP3:
$ CLOSE AEMINPUT
$ CLOSE TMPFILE2
$ COPY TEMP2.DAT IAEM1.DAT
$ PURGE
$ DEFINE/USER_MODE SYS$INPUT SYS$COMMAND
$ AEM
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT ""
$ OPEN IN OAEM1.DAT
$LOOP4:
$ READ/END_OF_FILE=ENDLOOP4 IN NAME
$ CHECK=F$EXTRACT(0,12,NAME)
$ IF CHECK .EQS. FIRST THEN LINE1 = F$EXTRACT(13,68,NAME)
$ IF CHECK .EQS. SECOND THEN LINE2 = F$EXTRACT(14,10,NAME)
$ IF CHECK .EQS. THIRD .AND. F$EXTRACT(82,1,NAME) .EQ. 1 -
  THEN LINE3 = F$EXTRACT(96,6,NAME)
$ IF CHECK .EQS. THIRD .AND. F$EXTRACT(82,1,NAME) .EQ. 2 -
  THEN LINE4 = F$EXTRACT(96,6,NAME)
$ IF CHECK .EQS. THIRD .AND. F$EXTRACT(82,1,NAME) .EQ. 3 -
  THEN LINE5 = F$EXTRACT(96,6,NAME)
$ IF CHECK .EQS. THIRD .AND. F$EXTRACT(82,1,NAME) .EQ. 4 -
  THEN LINE6 = F$EXTRACT(96,6,NAME)
$ IF CHECK .EQS. THIRD .AND. F$EXTRACT(82,1,NAME) .EQ. 5 -
  THEN LINE7 = F$EXTRACT(96,6,NAME)
$ GOTO LOOP4
$ENDLOOP4:
$ CLOSE IN
$ TEMPFRST = LINE1 + LINE2
$ FRSTLINE = F$EXTRACT(0,4,TEMPFRST) + F$EXTRACT(5,4,TEMPFRST) + -
F$EXTRACT(10,4,TEMPFRST)+F$EXTRACT(15,4,TEMPFRST)+F$EXTRACT(20,4,TEMPFRST) + -
F$EXTRACT(25,4,TEMPFRST)+F$EXTRACT(30,4,TEMPFRST)+F$EXTRACT(35,4,TEMPFRST) + -
F$EXTRACT(40,4,TEMPFRST)+F$EXTRACT(45,4,TEMPFRST)+F$EXTRACT(50,3,TEMPFRST) + -
F$EXTRACT(54,3,TEMPFRST)+F$EXTRACT(58,4,TEMPFRST)+F$EXTRACT(63,4,TEMPFRST) + -
F$EXTRACT(68,4,TEMPFRST)+F$EXTRACT(73,4,TEMPFRST)
$ LASTLINE = LINE3 + LINE4 + LINE5 + LINE6 + LINE7
$ OPEN/APPEND OUT REG.IN
$ WRITE OUT FRSTLINE
$ WRITE OUT LASTLINE
$ CLOSE OUT
$ IF COUNTER1 .EQ. REPS THEN GOTO FINISH
$ IF QUERY .NES. "Y" THEN GOTO LOOP1
```

```
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT ""
$ INQUIRE ANSWER "Do you wish to continue?  (y/n)"
$ IF ANSWER .EQS. "Y" THEN GOTO LOOP1
$FINISH:
$ OPEN RESULTS REG.IN
$ OPEN/APPEND STORAGE STORAGE.RES
$LOOP5:
$ READ/END_OF_FILE=ENDLOOP5 RESULTS RESLTREC
$ WRITE STORAGE RESLTREC
$ GOTO LOOP5
$ENDLOOP5:
$ CLOSE RESULTS
$ CLOSE STORAGE
$ COPY NULL.DAT REG.IN
$ RMAINING = REPS - COUNTER1
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT "You have ''RMAINING' repetitions left."
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT ""
$ EXIT
```

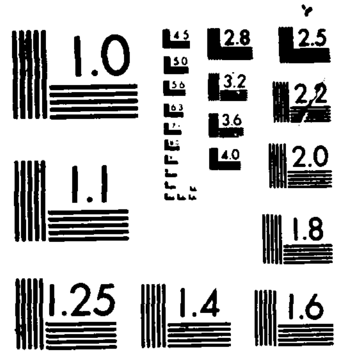
Appendix B: The IAEM1.DAT File

The IAEM.DAT file is the input file for executing the AEM. The AEM code for the AEM variables (weapons in this example -- i.e., target base, targeting strategies and priorities, and dependent variables) make up the main ingredients of the AEM input file.

```
$AGOGO
C
C Scenario Variables/Definitions
C
 TITLE=AEM Input File For Thesis Testing,
 WEP(1)=16,
 BTARGT(1,1)=MMII,MMIII,PKEEP,MICBM,POSC3,POSTRIC4,TRID5,B52GRV,B52SRM,B52CMC,
 BTARGT(11,1)=FB111SRM,FB111GRV,B1BGRV,B1BSRM,B1BALCM,ATB,
 BTARGT(1,2)=CIVIL,LOCAL,C3I,AFBASE,ICBM SILO,LCC,MICBM,NUKSTOR,SUBPTS,
 BTARGT(10,2)=SS-20,STORES,FACIL,DEPOS,NAVAL,FACTOR,POL,ENERGY,
 WCNAME(1)=BLUE_WEAPONS,BICBM,BSLBM,BBOMBER,BFAST,BSLOW,
 BLUE_WEAPONS=1-16,
 BICBM=1-4,
 BSLBM=5-7,
 BBOMBER=8-16,
 BFAST=1-7,
 BSLOW=8-16,
 TCNAME(1)=RED_TARGETS,RLC3I,RNUC,RCONV,RIND,RTIME_URGENT,
 RED_TARGETS=1-17,
 RLC3I=1-3,
 RNUC=4-10,
 RCONV=11-14,
 RIND=15-17,
 RTIME_URGENT=3-6,9,
C
C Print Variables:  Suppress most AEM print
C
 HEDGE PRINT=-2,-3,
 INPUT PRINT=-2,
 RESULT PRINT=-1,-2,-3,-4,-5,-6,-7,-8,-9,-10,-11,-12,-13,-14,
 STRIKE PRINT=-1,
 TARGET PRINT=-1,-2,-3,
 WEAPON PRINT=-1,-2,-3,-4,-5,-6,-7,-8,-9,-10,
C
C Weapon Variables (Side 1)
C
 BTARGT(1,1)=MMII,MMIII,PKEEP,MICBM,POSC3,POSTRIC4,TRID5,B52GRV,B52SRM
 JTYPE(1,1)=    1,    1,    1,    1,    2,        2,    2,      3,
```

AD-A185 247 AUTOMATING RESPONSE SURFACE METHODOLOGY FOR THE ARSENAL EXCHANGE MODEL(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. G E YIELDING 3/4

UNCLASSIFIED MAY 87 AFIT/GOR/ENS/86D-17 F/G 12/4 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
 TNUMB(1,1)=   77,  110,   45,   10,  190,     250,   10,     5,     5,    42,
 CEP(1,1)=     .2, .135,  .05,  .07,  .24,     .24,  .07,    .6,    .2,  .054,
 YIELD(1,1)=  1.2,  .34,  .34,  .34,  .05,      .1,  .34,     1,    .2,    .2,
 AV(1,1)=     .98,  .98,  .98,  .98,   .8,      .8,   .8,   .85,   .85,   .85,
 WHPC(1,1)=     1,    3,   10,    1,   10,       8,    8,     4,     4,    12,
 PLS(1,1)=     .6,   .6,   .8,   .8,   .9,    .925,  .95,    .9,    .9,    .9,
 PTP(1,1)=      1,    1,    1,    1,    1,       1,    1,    .6,    .6,   .71,
 RL(1,1)=     .85,   .9,   .9,  .85,   .8,      .8,  .85,    .8,    .8,   .75,
 BTARGT(11,1)=FB111SRM,FB111GRV,B1BGRV,B1BSRM,B1BALCM, ATB,
 JTYPE(11,1)=         3,        3,       3,       3,       3,   3,
 TNUMB(11,1)=        33,       33,      95,      95,      95, 102,
 CEP(11,1)=          .2,       .2,     .15,     .10,    .054, .05,
 YIELD(11,1)=        .2,        1,       1,      .2,      .2,  .5,
 AV(11,1)=          .85,      .85,     .85,     .85,     .85, .85,
 WHPC(11,1)=          4,        2,       4,       4,       8,  12,
 PLS(11,1)=          .9,       .9,     .95,     .95,     .95, .95,
 PTP(11,1)=         .67,      .67,     .84,     .84,     .84, .95,
 RL(11,1)=          .85,       .8,     .85,     .85,      .8, .85,
C
C Target Base/Variables (Side 2)
C
 BTARGT(1,2)=CIVIL,LOCAL, C3I,AFBASE,ICBM SILO, LCC,MICBM,NUKSTOR,SUBPTS,
 TCAT(1,2)=       3,     3,    2,     1,         1,    2,    1,       2,     2,
 TNUMB(1,2)=    140,   215,  450,    70,       640,  120,  150,      50,    20,
 R95(1,2)=      .28,  .245,    0,  .395,         0,    0,    0,       0,   .18,
 VAL(1,2)=        1,     1,    1,     1,         1,    1,    1,       1,     1,
 HARD(1,2)=    24Q0,  13P1,35P7,   10Q1,      52P8,39P0,    30,    39P5,  22P1,
 BTARGT(10,2)=SS-20,STORES,FACIL,DEPOS,NAVAL,FACTOR, POL,ENERGY,
 TCAT(10,2)=       1,     2,     2,     2,     2,     3,    3,     3,
 TNUMB(10,2)=    400,   430,   520,   550,   130,  1100,1300,   435,
 R95(10,2)=        0,     0,  .165,   .28,    .2,  .245,    0,   .11,
 VAL(10,2)=        1,     1,     1,     1,     1,     1,    1,     1,
 HARD(10,2)=    11P0,  31P6,  23Q0,  16Q0,  18P0,  14Q0,10P0,  18P0,
C
C Targeting Strategies:
C
C    Strategy One and Two:
C
 IMPACT FLAG=10,
 IMPACT(1,1)=16*3,
C
C    Strategy Three:
C
 CDM(1,2)=       .8,    .8,   .8,    .8,        .8,   .8,    .8,      .8,    .8,
 CDM(10,2)=       .8,    .7,    .7,    .7,    .7,    .6,   .6,    .6,
C
C    Strategy Four:
C
 ALLOW(3,2)=1-7,
 ALLOW(4,2)=1-7,
 ALLOW(5,2)=1-7,
 ALLOW(6,2)=1-7,
 ALLOW(7,2)=13,14,16,
```

```
 ALLOW(9,2)=1-7,
 ALLOW(10,2)=13,14,16,
C
C Value Accounts (The Dependent Variables)
C
 ATASK(1,1)=LC3I TRGTS,NUC TRGTS,CONV TRGTS,IND/ECON TRG,TOTAL TRGTS,
 ACCOUNT(1,1)=VALUE DEST ON (RLC3I) BY (ALL) MUST BE LE 1,
 ACCOUNT(2,1)=VALUE DEST ON (RNUC) BY (ALL) MUST BE LE 1,
 ACCOUNT(3,1)=VALUE DEST ON (RCONV) BY (ALL) MUST BE LE 1,
 ACCOUNT(4,1)=VALUE DEST ON (RIND) BY (ALL) MUST BE LE 1,
 ACCOUNT(5,1)=VALUE DEST ON (ALL) BY (ALL) MUST BE LE 1,
C
C Weapon Hedges (Targeting Priorities)
C
 TASK(1,1)=TIME URGENT,REST OF LC3I,REST OF NUC,CONV DAMAGE,IND DAMAGE,
 TASK(6,1)=ALL TARGETS,
 HEDGE(1,1)=TARGETS OF TYPE (RTIME_URGENT) BY (BICBM,BSLBM) MUST BE EQ 1,
 HEDGE(2,1)=TARGETS OF TYPE (1,2) BY (ALL) MUST BE EQ 1,
 HEDGE(3,1)=TARGETS OF TYPE (7,8,10) BY (ALL) MUST BE EQ 1,
 HEDGE(4,1)=TARGETS OF TYPE (RCONV) BY (ALL) MUST BE EQ 1,
 HEDGE(5,1)=TARGETS OF TYPE (RIND) BY (ALL) MUST BE EQ 1,
 HEDGE(6,1)=VALUE DEST ON (ALL) BY (ALL) MUST EQ 1,
 PRIORITY(1,1)=-1,-2,-3,-4,-5,-6,
$
```

Appendix F: The VARIBLES.DAT File

The VARIBLES.DAT file saves the design variable information entered with the INPTDATA subroutine if the user so desires. By line, the first column contains the name for the design variable; the second and third columns give the range, low and high respectively, of the design variable; and the fourth column shows how many AEM variables are controlled by the design variable.

MMII	0.00	450.00	1
MMIII	100.00	550.00	1
PKEEP	0.00	100.00	1
MICBM	0.00	200.00	1
POSC3	176.00	304.00	1
POSTRIC4	240.00	384.00	1
TRID5	0.00	336.00	1
B-52PEN	0.00	156.00	2
B-52CMC	40.00	156.00	1
FB111PEN	0.00	60.00	2
B1B	30.00	100.00	3
ATB	0.00	124.00	1

Appendix G: The CODEDSGN.IN File

The CODEDSGN.IN file is created from the INPTDATA subroutine. The first line gives the number of design variables processed through the INPTDATA subroutine. By column, the following lines list each design variable's low, mid, and high values for its range along with the number of AEM variables that each design variable controls.

```
12
   0.00  225.00  450.00 1
 100.00  325.00  550.00 1
   0.00   50.00  100.00 1
   0.00  100.00  200.00 1
 176.00  240.00  304.00 1
 240.00  312.00  384.00 1
   0.00  168.00  336.00 1
   0.00   78.00  156.00 2
  40.00   98.00  156.00 1
   0.00   30.00   60.00 2
  30.00   65.00  100.00 3
   0.00   62.00  124.00 1
```

Appendix H: The DECODE.IN File

The DECODE.IN file is created from the INPTDATA subroutine. The first column shows the names given each design variable. Columns two and three list each design variable's range, low and high values respectively. The fourth and fifth columns pertain to equation 2.3. The fourth column gives the midpoint of each design variable's range (the A variable of eqn 2.3), and the fifth column lists the value for the denominator of eqn 2.3 for each design variable (the B variable).

MMII	0.00	450.00	225.000000	225.000000
MMIII	100.00	550.00	325.000000	225.000000
PKEEP	0.00	100.00	50.000000	50.000000
MICBM	0.00	200.00	100.000000	100.000000
POSC3	176.00	304.00	240.000000	64.000000
POSTRIC4	240.00	384.00	312.000000	72.000000
TRID5	0.00	336.00	168.000000	168.000000
B-52PEN	0.00	156.00	78.000000	78.000000
B-52CMC	40.00	156.00	98.000000	58.000000
FB111PEN	0.00	60.00	30.000000	30.000000
B1B	30.00	100.00	65.000000	35.000000
ATB	0.00	124.00	62.000000	62.000000

Appendix I: The RGRSSION.IN File

The RGRSSION.IN file is created by the INPTDATA subroutine to be used later as input to the RGRSSION subroutine. The first line gives the number of design variables that is processed through the INPTDATA subroutine. The remaining lines list the names of the design variables.

```
12
MMII
MMIII
PKEEP
MICBM
POSC3
POSTRIC4
TRID5
B-52PEN
B-52CMC
FB111PEN
B1B
ATB
```

Appendix J: The AEM.DAT File

The AEM.DAT file is created from the CODEDSGN subroutine when selection U is picked from the main menu of the RSMed-AEM program. The AEM.DAT file is used with Tool Two (refer to Chapter Three) as the driver for the AEM executions. Tool Two reads in the AEM.DAT data two lines at a time. Each of the two lines replace the existing TNUMB values for the weapon variables (the sixteen AEM variables) in the IAEM1.DAT file (refer to Appendix E).

450.00	550.00	50.00	100.00	304.00	312.00	336.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
450.00	550.00	50.00	100.00	304.00	312.00	0.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
450.00	550.00	50.00	100.00	176.00	312.00	0.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
450.00	550.00	50.00	100.00	176.00	312.00	336.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
450.00	100.00	50.00	100.00	176.00	312.00	336.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
450.00	100.00	50.00	100.00	304.00	312.00	336.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
0.00	100.00	50.00	100.00	304.00	312.00	336.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
0.00	550.00	50.00	100.00	304.00	312.00	336.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
0.00	550.00	50.00	100.00	304.00	312.00	0.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
0.00	550.00	50.00	100.00	176.00	312.00	0.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
0.00	550.00	50.00	100.00	176.00	312.00	336.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
0.00	100.00	50.00	100.00	176.00	312.00	336.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
0.00	100.00	50.00	100.00	176.00	312.00	0.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
0.00	100.00	50.00	100.00	304.00	312.00	0.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
450.00	100.00	50.00	100.00	304.00	312.00	0.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
450.00	100.00	50.00	100.00	176.00	312.00	0.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				

225.00	550.00	100.00	100.00	240.00	384.00	168.00	156.00	156.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	550.00	100.00	100.00	240.00	384.00	168.00	0.00	0.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	550.00	100.00	100.00	240.00	240.00	168.00	0.00	0.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	550.00	100.00	100.00	240.00	240.00	168.00	156.00	156.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	550.00	0.00	100.00	240.00	240.00	168.00	156.00	156.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	550.00	0.00	100.00	240.00	384.00	168.00	156.00	156.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	100.00	0.00	100.00	240.00	384.00	168.00	156.00	156.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	100.00	100.00	100.00	240.00	384.00	168.00	156.00	156.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	100.00	100.00	100.00	240.00	384.00	168.00	0.00	0.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	100.00	100.00	100.00	240.00	240.00	168.00	0.00	0.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	100.00	100.00	100.00	240.00	240.00	168.00	156.00	156.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	100.00	0.00	100.00	240.00	240.00	168.00	156.00	156.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	100.00	0.00	100.00	240.00	240.00	168.00	0.00	0.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	100.00	0.00	100.00	240.00	384.00	168.00	0.00	0.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	550.00	0.00	100.00	240.00	384.00	168.00	0.00	0.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	550.00	0.00	100.00	240.00	240.00	168.00	0.00	0.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	325.00	100.00	200.00	240.00	312.00	336.00	78.00	78.00	156.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	325.00	100.00	200.00	240.00	312.00	336.00	78.00	78.00	40.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	325.00	100.00	200.00	240.00	312.00	0.00	78.00	78.00	40.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	325.00	100.00	200.00	240.00	312.00	0.00	78.00	78.00	156.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	325.00	100.00	0.00	240.00	312.00	0.00	78.00	78.00	156.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	325.00	100.00	0.00	240.00	312.00	336.00	78.00	78.00	156.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	325.00	0.00	0.00	240.00	312.00	336.00	78.00	78.00	156.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	325.00	0.00	200.00	240.00	312.00	336.00	78.00	78.00	156.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	325.00	0.00	200.00	240.00	312.00	336.00	78.00	78.00	40.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	325.00	0.00	200.00	240.00	312.00	0.00	78.00	78.00	40.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	325.00	0.00	200.00	240.00	312.00	0.00	78.00	78.00	156.00

30.00	30.00	65.00	65.00	65.00	62.00				
225.00	325.00	0.00	0.00	240.00	312.00	0.00	78.00	78.00	156.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	325.00	0.00	0.00	240.00	312.00	0.00	78.00	78.00	40.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	325.00	0.00	0.00	240.00	312.00	336.00	78.00	78.00	40.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	325.00	100.00	0.00	240.00	312.00	336.00	78.00	78.00	40.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	325.00	100.00	0.00	240.00	312.00	0.00	78.00	78.00	40.00
30.00	30.00	65.00	65.00	65.00	62.00				
225.00	325.00	50.00	200.00	304.00	312.00	168.00	156.00	156.00	98.00
60.00	60.00	65.00	65.00	65.00	62.00				
225.00	325.00	50.00	200.00	304.00	312.00	168.00	156.00	156.00	98.00
0.00	0.00	65.00	65.00	65.00	62.00				
225.00	325.00	50.00	200.00	304.00	312.00	168.00	0.00	0.00	98.00
0.00	0.00	65.00	65.00	65.00	62.00				
225.00	325.00	50.00	200.00	304.00	312.00	168.00	0.00	0.00	98.00
60.00	60.00	65.00	65.00	65.00	62.00				
225.00	325.00	50.00	200.00	176.00	312.00	168.00	0.00	0.00	98.00
60.00	60.00	65.00	65.00	65.00	62.00				
225.00	325.00	50.00	200.00	176.00	312.00	168.00	156.00	156.00	98.00
60.00	60.00	65.00	65.00	65.00	62.00				
225.00	325.00	50.00	0.00	176.00	312.00	168.00	156.00	156.00	98.00
60.00	60.00	65.00	65.00	65.00	62.00				
225.00	325.00	50.00	0.00	304.00	312.00	168.00	156.00	156.00	98.00
60.00	60.00	65.00	65.00	65.00	62.00				
225.00	325.00	50.00	0.00	304.00	312.00	168.00	156.00	156.00	98.00
0.00	0.00	65.00	65.00	65.00	62.00				
225.00	325.00	50.00	0.00	304.00	312.00	168.00	0.00	0.00	98.00
0.00	0.00	65.00	65.00	65.00	62.00				
225.00	325.00	50.00	0.00	304.00	312.00	168.00	0.00	0.00	98.00
60.00	60.00	65.00	65.00	65.00	62.00				
225.00	325.00	50.00	0.00	176.00	312.00	168.00	0.00	0.00	98.00
60.00	60.00	65.00	65.00	65.00	62.00				
225.00	325.00	50.00	0.00	176.00	312.00	168.00	0.00	0.00	98.00
0.00	0.00	65.00	65.00	65.00	62.00				
225.00	325.00	50.00	0.00	176.00	312.00	168.00	156.00	156.00	98.00
0.00	0.00	65.00	65.00	65.00	62.00				
225.00	325.00	50.00	200.00	176.00	312.00	168.00	156.00	156.00	98.00
0.00	0.00	65.00	65.00	65.00	62.00				
225.00	325.00	50.00	200.00	176.00	312.00	168.00	0.00	0.00	98.00
0.00	0.00	65.00	65.00	65.00	62.00				
225.00	325.00	50.00	100.00	304.00	384.00	168.00	78.00	78.00	156.00
30.00	30.00	100.00	100.00	100.00	62.00				
225.00	325.00	50.00	100.00	304.00	384.00	168.00	78.00	78.00	156.00
30.00	30.00	30.00	30.00	30.00	62.00				
225.00	325.00	50.00	100.00	304.00	384.00	168.00	78.00	78.00	40.00
30.00	30.00	30.00	30.00	30.00	62.00				
225.00	325.00	50.00	100.00	304.00	384.00	168.00	78.00	78.00	40.00
30.00	30.00	100.00	100.00	100.00	62.00				
225.00	325.00	50.00	100.00	304.00	240.00	168.00	78.00	78.00	40.00
30.00	30.00	100.00	100.00	100.00	62.00				

225.00	325.00	50.00	100.00	304.00	240.00	168.00	78.00	78.00	156.00
30.00	30.00	100.00	100.00	100.00	62.00				
225.00	325.00	50.00	100.00	176.00	240.00	168.00	78.00	78.00	156.00
30.00	30.00	100.00	100.00	100.00	62.00				
225.00	325.00	50.00	100.00	176.00	384.00	168.00	78.00	78.00	156.00
30.00	30.00	100.00	100.00	100.00	62.00				
225.00	325.00	50.00	100.00	176.00	384.00	168.00	78.00	78.00	156.00
30.00	30.00	30.00	30.00	30.00	62.00				
225.00	325.00	50.00	100.00	176.00	384.00	168.00	78.00	78.00	40.00
30.00	30.00	30.00	30.00	30.00	62.00				
225.00	325.00	50.00	100.00	176.00	384.00	168.00	78.00	78.00	40.00
30.00	30.00	100.00	100.00	100.00	62.00				
225.00	325.00	50.00	100.00	176.00	240.00	168.00	78.00	78.00	40.00
30.00	30.00	100.00	100.00	100.00	62.00				
225.00	325.00	50.00	100.00	176.00	240.00	168.00	78.00	78.00	40.00
30.00	30.00	30.00	30.00	30.00	62.00				
225.00	325.00	50.00	100.00	176.00	240.00	168.00	78.00	78.00	156.00
30.00	30.00	30.00	30.00	30.00	62.00				
225.00	325.00	50.00	100.00	304.00	240.00	168.00	78.00	78.00	156.00
30.00	30.00	30.00	30.00	30.00	62.00				
225.00	325.00	50.00	100.00	304.00	240.00	168.00	78.00	78.00	40.00
30.00	30.00	30.00	30.00	30.00	62.00				
225.00	325.00	50.00	100.00	240.00	384.00	336.00	78.00	78.00	98.00
60.00	60.00	65.00	65.00	65.00	124.00				
225.00	325.00	50.00	100.00	240.00	384.00	336.00	78.00	78.00	98.00
60.00	60.00	65.00	65.00	65.00	0.00				
225.00	325.00	50.00	100.00	240.00	384.00	336.00	78.00	78.00	98.00
0.00	0.00	65.00	65.00	65.00	0.00				
225.00	325.00	50.00	100.00	240.00	384.00	336.00	78.00	78.00	98.00
0.00	0.00	65.00	65.00	65.00	124.00				
225.00	325.00	50.00	100.00	240.00	384.00	0.00	78.00	78.00	98.00
0.00	0.00	65.00	65.00	65.00	124.00				
225.00	325.00	50.00	100.00	240.00	384.00	0.00	78.00	78.00	98.00
60.00	60.00	65.00	65.00	65.00	124.00				
225.00	325.00	50.00	100.00	240.00	240.00	0.00	78.00	78.00	98.00
60.00	60.00	65.00	65.00	65.00	124.00				
225.00	325.00	50.00	100.00	240.00	240.00	336.00	78.00	78.00	98.00
60.00	60.00	65.00	65.00	65.00	124.00				
225.00	325.00	50.00	100.00	240.00	240.00	336.00	78.00	78.00	98.00
60.00	60.00	65.00	65.00	65.00	0.00				
225.00	325.00	50.00	100.00	240.00	240.00	336.00	78.00	78.00	98.00
0.00	0.00	65.00	65.00	65.00	0.00				
225.00	325.00	50.00	100.00	240.00	240.00	336.00	78.00	78.00	98.00
0.00	0.00	65.00	65.00	65.00	124.00				
225.00	325.00	50.00	100.00	240.00	240.00	0.00	78.00	78.00	98.00
0.00	0.00	65.00	65.00	65.00	124.00				
225.00	325.00	50.00	100.00	240.00	240.00	0.00	78.00	78.00	98.00
0.00	0.00	65.00	65.00	65.00	0.00				
225.00	325.00	50.00	100.00	240.00	240.00	0.00	78.00	78.00	98.00
60.00	60.00	65.00	65.00	65.00	0.00				
225.00	325.00	50.00	100.00	240.00	384.00	0.00	78.00	78.00	98.00
60.00	60.00	65.00	65.00	65.00	0.00				
225.00	325.00	50.00	100.00	240.00	384.00	0.00	78.00	78.00	98.00

0.00	0.00	65.00	65.00	65.00	0.00				
450.00	325.00	50.00	100.00	240.00	312.00	336.00	156.00	156.00	98.00
30.00	30.00	100.00	100.00	100.00	62.00				
450.00	325.00	50.00	100.00	240.00	312.00	336.00	156.00	156.00	98.00
30.00	30.00	30.00	30.00	30.00	62.00				
450.00	325.00	50.00	100.00	240.00	312.00	336.00	0.00	0.00	98.00
30.00	30.00	30.00	30.00	30.00	62.00				
450.00	325.00	50.00	100.00	240.00	312.00	336.00	0.00	0.00	98.00
30.00	30.00	100.00	100.00	100.00	62.00				
450.00	325.00	50.00	100.00	240.00	312.00	0.00	0.00	0.00	98.00
30.00	30.00	100.00	100.00	100.00	62.00				
450.00	325.00	50.00	100.00	240.00	312.00	0.00	156.00	156.00	98.00
30.00	30.00	100.00	100.00	100.00	62.00				
0.00	325.00	50.00	100.00	240.00	312.00	0.00	156.00	156.00	98.00
30.00	30.00	100.00	100.00	100.00	62.00				
0.00	325.00	50.00	100.00	240.00	312.00	336.00	156.00	156.00	98.00
30.00	30.00	100.00	100.00	100.00	62.00				
0.00	325.00	50.00	100.00	240.00	312.00	336.00	156.00	156.00	98.00
30.00	30.00	30.00	30.00	30.00	62.00				
0.00	325.00	50.00	100.00	240.00	312.00	336.00	0.00	0.00	98.00
30.00	30.00	30.00	30.00	30.00	62.00				
0.00	325.00	50.00	100.00	240.00	312.00	336.00	0.00	0.00	98.00
30.00	30.00	100.00	100.00	100.00	62.00				
0.00	325.00	50.00	100.00	240.00	312.00	0.00	0.00	0.00	98.00
30.00	30.00	100.00	100.00	100.00	62.00				
0.00	325.00	50.00	100.00	240.00	312.00	0.00	0.00	0.00	98.00
30.00	30.00	30.00	30.00	30.00	62.00				
0.00	325.00	50.00	100.00	240.00	312.00	0.00	156.00	156.00	98.00
30.00	30.00	30.00	30.00	30.00	62.00				
450.00	325.00	50.00	100.00	240.00	312.00	0.00	156.00	156.00	98.00
30.00	30.00	30.00	30.00	30.00	62.00				
450.00	325.00	50.00	100.00	240.00	312.00	0.00	0.00	0.00	98.00
30.00	30.00	30.00	30.00	30.00	62.00				
225.00	550.00	50.00	100.00	240.00	312.00	168.00	156.00	156.00	156.00
30.00	30.00	65.00	65.00	65.00	124.00				
225.00	550.00	50.00	100.00	240.00	312.00	168.00	156.00	156.00	156.00
30.00	30.00	65.00	65.00	65.00	0.00				
225.00	550.00	50.00	100.00	240.00	312.00	168.00	156.00	156.00	40.00
30.00	30.00	65.00	65.00	65.00	0.00				
225.00	550.00	50.00	100.00	240.00	312.00	168.00	156.00	156.00	40.00
30.00	30.00	65.00	65.00	65.00	124.00				
225.00	550.00	50.00	100.00	240.00	312.00	168.00	0.00	0.00	40.00
30.00	30.00	65.00	65.00	65.00	124.00				
225.00	550.00	50.00	100.00	240.00	312.00	168.00	0.00	0.00	156.00
30.00	30.00	65.00	65.00	65.00	124.00				
225.00	100.00	50.00	100.00	240.00	312.00	168.00	0.00	0.00	156.00
30.00	30.00	65.00	65.00	65.00	124.00				
225.00	100.00	50.00	100.00	240.00	312.00	168.00	156.00	156.00	156.00
30.00	30.00	65.00	65.00	65.00	124.00				
225.00	100.00	50.00	100.00	240.00	312.00	168.00	156.00	156.00	156.00
30.00	30.00	65.00	65.00	65.00	0.00				
225.00	100.00	50.00	100.00	240.00	312.00	168.00	156.00	156.00	40.00
30.00	30.00	65.00	65.00	65.00	0.00				

225.00	100.00	50.00	100.00	240.00	312.00	168.00	156.00	156.00	40.00
30.00	30.00	65.00	65.00	65.00	124.00				
225.00	100.00	50.00	100.00	240.00	312.00	168.00	0.00	0.00	40.00
30.00	30.00	65.00	65.00	65.00	124.00				
225.00	100.00	50.00	100.00	240.00	312.00	168.00	0.00	0.00	40.00
30.00	30.00	65.00	65.00	65.00	0.00				
225.00	100.00	50.00	100.00	240.00	312.00	168.00	0.00	0.00	156.00
30.00	30.00	65.00	65.00	65.00	0.00				
225.00	550.00	50.00	100.00	240.00	312.00	168.00	0.00	0.00	156.00
30.00	30.00	65.00	65.00	65.00	0.00				
225.00	550.00	50.00	100.00	240.00	312.00	168.00	0.00	0.00	40.00
30.00	30.00	65.00	65.00	65.00	0.00				
450.00	325.00	100.00	100.00	240.00	312.00	168.00	78.00	78.00	156.00
60.00	60.00	65.00	65.00	65.00	62.00				
450.00	325.00	100.00	100.00	240.00	312.00	168.00	78.00	78.00	156.00
0.00	0.00	65.00	65.00	65.00	62.00				
450.00	325.00	100.00	100.00	240.00	312.00	168.00	78.00	78.00	40.00
0.00	0.00	65.00	65.00	65.00	62.00				
450.00	325.00	100.00	100.00	240.00	312.00	168.00	78.00	78.00	40.00
60.00	60.00	65.00	65.00	65.00	62.00				
450.00	325.00	0.00	100.00	240.00	312.00	168.00	78.00	78.00	40.00
60.00	60.00	65.00	65.00	65.00	62.00				
450.00	325.00	0.00	100.00	240.00	312.00	168.00	78.00	78.00	156.00
60.00	60.00	65.00	65.00	65.00	62.00				
0.00	325.00	0.00	100.00	240.00	312.00	168.00	78.00	78.00	156.00
60.00	60.00	65.00	65.00	65.00	62.00				
0.00	325.00	100.00	100.00	240.00	312.00	168.00	78.00	78.00	156.00
60.00	60.00	65.00	65.00	65.00	62.00				
0.00	325.00	100.00	100.00	240.00	312.00	168.00	78.00	78.00	156.00
0.00	0.00	65.00	65.00	65.00	62.00				
0.00	325.00	100.00	100.00	240.00	312.00	168.00	78.00	78.00	40.00
0.00	0.00	65.00	65.00	65.00	62.00				
0.00	325.00	100.00	100.00	240.00	312.00	168.00	78.00	78.00	40.00
60.00	60.00	65.00	65.00	65.00	62.00				
0.00	325.00	0.00	100.00	240.00	312.00	168.00	78.00	78.00	40.00
60.00	60.00	65.00	65.00	65.00	62.00				
0.00	325.00	0.00	100.00	240.00	312.00	168.00	78.00	78.00	40.00
0.00	0.00	65.00	65.00	65.00	62.00				
0.00	325.00	0.00	100.00	240.00	312.00	168.00	78.00	78.00	156.00
0.00	0.00	65.00	65.00	65.00	62.00				
450.00	325.00	0.00	100.00	240.00	312.00	168.00	78.00	78.00	156.00
0.00	0.00	65.00	65.00	65.00	62.00				
450.00	325.00	0.00	100.00	240.00	312.00	168.00	78.00	78.00	40.00
0.00	0.00	65.00	65.00	65.00	62.00				
225.00	550.00	50.00	200.00	240.00	312.00	168.00	78.00	78.00	98.00
60.00	60.00	100.00	100.00	100.00	62.00				
225.00	550.00	50.00	200.00	240.00	312.00	168.00	78.00	78.00	98.00
60.00	60.00	30.00	30.00	30.00	62.00				
225.00	550.00	50.00	200.00	240.00	312.00	168.00	78.00	78.00	98.00
0.00	0.00	30.00	30.00	30.00	62.00				
225.00	550.00	50.00	200.00	240.00	312.00	168.00	78.00	78.00	98.00
0.00	0.00	100.00	100.00	100.00	62.00				
225.00	550.00	50.00	0.00	240.00	312.00	168.00	78.00	78.00	98.00

0.00	0.00	100.00	100.00	100.00	62.00				
225.00	550.00	50.00	0.00	240.00	312.00	168.00	78.00	78.00	98.00
60.00	60.00	100.00	100.00	100.00	62.00				
225.00	100.00	50.00	0.00	240.00	312.00	168.00	78.00	78.00	98.00
60.00	60.00	100.00	100.00	100.00	62.00				
225.00	100.00	50.00	200.00	240.00	312.00	168.00	78.00	78.00	98.00
60.00	60.00	100.00	100.00	100.00	62.00				
225.00	100.00	50.00	200.00	240.00	312.00	168.00	78.00	78.00	98.00
60.00	60.00	30.00	30.00	30.00	62.00				
225.00	100.00	50.00	200.00	240.00	312.00	168.00	78.00	78.00	98.00
0.00	0.00	30.00	30.00	30.00	62.00				
225.00	100.00	50.00	200.00	240.00	312.00	168.00	78.00	78.00	98.00
0.00	0.00	100.00	100.00	100.00	62.00				
225.00	100.00	50.00	0.00	240.00	312.00	168.00	78.00	78.00	98.00
0.00	0.00	100.00	100.00	100.00	62.00				
225.00	100.00	50.00	0.00	240.00	312.00	168.00	78.00	78.00	98.00
0.00	0.00	30.00	30.00	30.00	62.00				
225.00	100.00	50.00	0.00	240.00	312.00	168.00	78.00	78.00	98.00
60.00	60.00	30.00	30.00	30.00	62.00				
225.00	550.00	50.00	0.00	240.00	312.00	168.00	78.00	78.00	98.00
60.00	60.00	30.00	30.00	30.00	62.00				
225.00	550.00	50.00	0.00	240.00	312.00	168.00	78.00	78.00	98.00
0.00	0.00	30.00	30.00	30.00	62.00				
225.00	325.00	100.00	100.00	304.00	312.00	168.00	78.00	78.00	98.00
30.00	30.00	100.00	100.00	100.00	124.00				
225.00	325.00	100.00	100.00	304.00	312.00	168.00	78.00	78.00	98.00
30.00	30.00	100.00	100.00	100.00	0.00				
225.00	325.00	100.00	100.00	304.00	312.00	168.00	78.00	78.00	98.00
30.00	30.00	30.00	30.00	30.00	0.00				
225.00	325.00	100.00	100.00	304.00	312.00	168.00	78.00	78.00	98.00
30.00	30.00	30.00	30.00	30.00	124.00				
225.00	325.00	100.00	100.00	176.00	312.00	168.00	78.00	78.00	98.00
30.00	30.00	30.00	30.00	30.00	124.00				
225.00	325.00	100.00	100.00	176.00	312.00	168.00	78.00	78.00	98.00
30.00	30.00	100.00	100.00	100.00	124.00				
225.00	325.00	0.00	100.00	176.00	312.00	168.00	78.00	78.00	98.00
30.00	30.00	100.00	100.00	100.00	124.00				
225.00	325.00	0.00	100.00	304.00	312.00	168.00	78.00	78.00	98.00
30.00	30.00	100.00	100.00	100.00	124.00				
225.00	325.00	0.00	100.00	304.00	312.00	168.00	78.00	78.00	98.00
30.00	30.00	100.00	100.00	100.00	0.00				
225.00	325.00	0.00	100.00	304.00	312.00	168.00	78.00	78.00	98.00
30.00	30.00	30.00	30.00	30.00	0.00				
225.00	325.00	0.00	100.00	304.00	312.00	168.00	78.00	78.00	98.00
30.00	30.00	30.00	30.00	30.00	124.00				
225.00	325.00	0.00	100.00	176.00	312.00	168.00	78.00	78.00	98.00
30.00	30.00	30.00	30.00	30.00	124.00				
225.00	325.00	0.00	100.00	176.00	312.00	168.00	78.00	78.00	98.00
30.00	30.00	30.00	30.00	30.00	0.00				
225.00	325.00	0.00	100.00	176.00	312.00	168.00	78.00	78.00	98.00
30.00	30.00	100.00	100.00	100.00	0.00				
225.00	325.00	100.00	100.00	176.00	312.00	168.00	78.00	78.00	98.00
30.00	30.00	100.00	100.00	100.00	0.00				

225.00	325.00	100.00	100.00	176.00	312.00	168.00	78.00	78.00	98.00
30.00	30.00	30.00	30.00	30.00	0.00				
450.00	325.00	50.00	200.00	240.00	384.00	168.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	124.00				
450.00	325.00	50.00	200.00	240.00	384.00	168.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	0.00				
450.00	325.00	50.00	200.00	240.00	240.00	168.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	0.00				
450.00	325.00	50.00	200.00	240.00	240.00	168.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	124.00				
450.00	325.00	50.00	0.00	240.00	240.00	168.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	124.00				
450.00	325.00	50.00	0.00	240.00	384.00	168.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	124.00				
0.00	325.00	50.00	0.00	240.00	384.00	168.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	124.00				
0.00	325.00	50.00	200.00	240.00	384.00	168.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	124.00				
0.00	325.00	50.00	200.00	240.00	384.00	168.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	0.00				
0.00	325.00	50.00	200.00	240.00	240.00	168.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	0.00				
0.00	325.00	50.00	200.00	240.00	240.00	168.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	124.00				
0.00	325.00	50.00	0.00	240.00	240.00	168.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	124.00				
0.00	325.00	50.00	0.00	240.00	240.00	168.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	0.00				
0.00	325.00	50.00	0.00	240.00	384.00	168.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	0.00				
450.00	325.00	50.00	0.00	240.00	384.00	168.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	0.00				
450.00	325.00	50.00	0.00	240.00	240.00	168.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	0.00				
225.00	325.00	50.00	100.00	240.00	312.00	168.00	78.00	78.00	98.00
30.00	30.00	65.00	65.00	65.00	62.00				

Appendix K: The OAEM1.DAT FILE

Each execution of the IAEM1.DAT file (Appendix E) produces the AEM output file -- OAEM1.DAT. It is this file that Tool Two (Appendix D) extracts the dependent variable information from. (Note: The AEM outputs the OAEM1.DAT file in a local printer format. Therefore, the file is 132 characters wide. Consequently, copying 132 columns into 72 columns, for this appendix, produces some difficulty in text appearance and text reading. Just remember, sometimes two lines are required to represent one line of actual OAEM1.DAT data.)

The OAEM1.DAT file shown here in this appendix is not indicative of the amount of print the AEM will generate. Most of the output print has been shut off through the AEM print flags. Only the print used in the RSMed-AEM program test is shown.

```
INUT FILE IS IAEM1                                    OUTPUT FILE NAME IS OAE
M1
AEM HEDGE        AUGUST 1985
************************************************** INPUT DATA ****************
*********************************
(I-1)

                          ESTIMATES OF DATA FOR BOTH SIDES
$AGOGO
C
C Scenario Variables
C
```

```
TITLE=TACT 7.71 Strategic Force Analysis Projects,
WEP(1)=16,
BTARGT(1,1)=MMII,MMIII,PKEEP,MICBM,POSC3,POSTRIC4,TRID5,B52GRV,B52SRM,B52CMC,
BTARGT(11,1)=FB111SRM,FB111GRV,B1BGRV,B1BSRM,B1BALCM,ATB,
BTARGT(1,2)=CIVIL,LOCAL,C3I,AFBASE,ICBM SILO,LCC,MICBM,NUKSTOR,SUBPTS,
BTARGT(10,2)=SS-20,STORES,FACIL,DEPOS,NAVAL,FACTOR,POL,ENERGY,
WCNAME(1)=BLUE_WEAPONS,BICBM,BSLBM,BBOMBER,BFAST,BSLOW,
BLUE_WEAPONS=1-16,
BICBM=1-4,
BSLBM=5-7,
BBOMBER=8-16,
BFAST=1-7,
BSLOW=8-16,
TCNAME(1)=RED_TARGETS,RLC3I,RNUC,RCONV,RIND,RTIME_URGENT,
RED_TARGETS=1-17,
RLC3I=1-3,
RNUC=4-10,
RCONV=11-14,
RIND=15-17,
RTIME_URGENT=3-6,9,
 IMPACT FLAG=10,
 IMPACT(1,1)=16*3,
C
C Print Variables
C
 HEDGE PRINT=-2,-3,
 INPUT PRINT=-2,
 RESULT PRINT=-1,-2,-3,-4,-5,-6,-7,-8,-9,-10,-11,-12,-13,-14,
 STRIKE PRINT=-1,
 TARGET PRINT=-1,-2,-3,
 WEAPON PRINT=-1,-2,-3,-4,-5,-6,-7,-8,-9,-10,
C
C Weapon Variables (Side 1)
C
 BTARGT(1,1)=MMII,MMIII,PKEEP,MICBM,POSC3,POSTRIC4,TRID5,B52GRV,B52SRM,B52CMC,
 JTYPE(1,1)=     1,     1,     1,     1,     2,        2,     2,     3,     3,     3,
 TNUMB(1,1)=  77, 110,   45,   10, 190, 250,   10,    5,    5,  42, 33, 33,  95,  95,
 TNUMB(15,1)=  95, 102,
 CEP(1,1)=      .2, .135,  .05,  .07,  .24,      .24,  .07,     .6,     .2,  .054,
 YIELD(1,1)=  1.2,  .34,  .34,  .34,  .05,       .1,  .34,      1,     .2,    .2,
 AV(1,1)=      .98,  .98,  .98,  .98,   .8,       .8,   .8,    .85,    .85,   .85,
 WHPC(1,1)=      1,    3,   10,    1,   10,        8,    8,      4,      4,    12,
 PLS(1,1)=      .6,   .6,   .8,   .8,   .9,     .925,  .95,     .9,     .9,    .9,
 PTP(1,1)=       1,    1,    1,    1,    1,        1,    1,     .6,     .6,   .71,
 RL(1,1)=      .85,   .9,   .9,  .85,   .8,       .8,  .85,     .8,     .8,   .75,
 BTARGT(11,1)=FB111SRM,FB111GRV,B1BGRV,B1BSRM,B1BALCM, ATB,
 JTYPE(11,1)=         3,         3,      3,      3,        3,   3,
 CEP(11,1)=          .2,        .2,    .15,    .10,     .054, .05,
 YIELD(11,1)=        .2,         1,      1,     .2,       .2,  .5,
 AV(11,1)=          .85,       .85,    .85,    .85,      .85, .85,
 WHPC(11,1)=          4,         2,      4,      4,        8,  12,
 PLS(11,1)=          .9,        .9,    .95,    .95,      .95, .95,
 PTP(11,1)=         .67,       .67,    .84,    .84,      .84, .95,
```

```
RL(11,1)=         .85,      .8,   .85,   .85,     .8, .85,
C
C Target Variables (Side 2)
C
 BTARGT(1,2)=CIVIL,LOCAL, C3I,AFBASE,ICBM SILO, LCC,MICBM,NUKSTOR,SUBPTS,
 TCAT(1,2)=       3,    3,   2,      1,          1,   2,    1,       2,     2,
 THUMB(1,2)=   140,  215, 450,     70,        640, 120,  150,      50,    20,
 R95(1,2)=      .28, .245,   0,   .395,          0,   0,    0,       0,   .18,
 VAL(1,2)=        1,    1,   1,      1,          1,   1,    1,       1,     1,
 HARD(1,2)=   24Q0, 13P1,35P7,   10Q1,       52P8,39P0,   30,    39P5,  22P1,
 CDM(1,2)=       .8,   .8,  .8,     .8,         .8,  .8,   .8,      .8,    .8,
 BTARGT(10,2)=SS-20,STORES,FACIL,DEPOS,NAVAL,FACTOR, POL,ENERGY,
 TCAT(10,2)=      1,     2,    2,    2,    2,     3,   3,     3,
 THUMB(10,2)=   400,   430,  520,  550,  130,  1100,1300,   435,
 R95(10,2)=       0,     0, .165,  .28,   .2,  .245,   0,   .11,
 VAL(10,2)=       1,     1,    1,    1,    1,     1,   1,     1,
 HARD(10,2)=   11P0,  31P6, 23Q0, 16Q0, 18P0,  14Q0,10P0,  18P0,
 CDM(10,2)=      .8,    .7,   .7,   .7,   .7,    .6,  .6,    .6,
 ALLOW(3,2)=1-7,
 ALLOW(4,2)=1-7,
 ALLOW(5,2)=1-7,
 ALLOW(6,2)=1-7,
 ALLOW(7,2)=13,14,16,
 ALLOW(9,2)=1-7,
 ALLOW(10,2)=13,14,16,
C
C Value Accounts
C
 ATASK(1,1)=LC3I TRGTS,NUC TRGTS,CONV TRGTS,IND/ECON TRG,TOTAL TRGTS,
 ACCOUNT(1,1)=VALUE DEST ON (RLC3I) BY (ALL) MUST BE LE 1,
 ACCOUNT(2,1)=VALUE DEST ON (RNUC) BY (ALL) MUST BE LE 1,
 ACCOUNT(3,1)=VALUE DEST ON (RCONV) BY (ALL) MUST BE LE 1,
 ACCOUNT(4,1)=VALUE DEST ON (RIND) BY (ALL) MUST BE LE 1,
 ACCOUNT(5,1)=VALUE DEST ON (ALL) BY (ALL) MUST BE LE 1,
C
C Weapon Hedges
C
 TASK(1,1)=TIME URGENT,REST OF LC3I,REST OF NUC,CONV DAMAGE,IND DAMAGE,
 TASK(6,1)=ALL TARGETS,
 HEDGE(1,1)=TARGETS OF TYPE (RTIME_URGENT) BY (BICBM,BSLBM) MUST BE EQ 1,
 HEDGE(2,1)=TARGETS OF TYPE (1,2) BY (ALL) MUST BE EQ 1,
 HEDGE(3,1)=TARGETS OF TYPE (7,8,10) BY (ALL) MUST BE EQ 1,
 HEDGE(4,1)=TARGETS OF TYPE (RCONV) BY (ALL) MUST BE EQ 1,
 HEDGE(5,1)=TARGETS OF TYPE (RIND) BY (ALL) MUST BE EQ 1,
 HEDGE(6,1)=VALUE DEST ON (ALL) BY (ALL) MUST EQ 1,
 PRIORITY(1,1)=-1,-2,-3,-4,-5,-6,
$
```

```
LP NUMBER   STRATS  TOTAL STRATS PIVOTS PAYOFF EXIT CRITERION
      1        21        128        27    -35195.93  0.000000
      2        29        146        44    -28314.33 -0.008136
AM MODIFYING HEDGE     1 ORIGINAL GOAL=   1299.35 ACHIEVED=     512.35
NOW WORKING ON PRIORITY NUMBER  2
NOW WORKING ON PRIORITY NUMBER  3
NOW WORKING ON PRIORITY NUMBER  4
NOW WORKING ON PRIORITY NUMBER  5
      3        29        146         2    -13117.34 -0.007149
      4        29        143         4    -12377.90 -0.006065
AM MODIFYING HEDGE     5 ORIGINAL GOAL=   2833.58 ACHIEVED=    1971.35
NOW WORKING ON PRIORITY NUMBER  6
      5        30        143         2    -17915.95 -0.001779
      6        30        143         2    -17915.74  0.000000
AM MODIFYING HEDGE     6 ORIGINAL GOAL=   6716.64 ACHIEVED=    3980.93
NOW WORKING ON PRIORITY NUMBER  0
      7        30        144         2      3981.50 -0.000977
      8        30        148         0      3980.93  0.000000
```

```
****************************************************************************************
************************************************
                    TACT 7.71 Strategic Force Analysis Projects

                         IAEM1,

(H-1)                                                          HEDGE CONDITION DATA


                 LEVEL      PERCENT
   CONDITION ACHIEVED      ACHIEVED     LAMBDA      PREFERENCE   PRIORITY    TASK

*       1       0.39         39.43         0.00          -8.00         -1    TIME UR
GENT            *** NOT ACHIEVED
  NO. OF TGTS OF TYPE (3-6,9) ATTACKED BY WPNS OF TYPE (1-7) MUST BE EQ  1,

*       2       1.00        100.00         0.00          -8.00         -2    REST OF
 LC3I
  NO. OF TGTS OF TYPE (1-2) ATTACKED BY WPNS OF TYPE (ALL    ) MUST BE EQ  1,

*       3       1.00        100.00         0.00          -8.00         -3    REST OF
 NUC
  NO. OF TGTS OF TYPE (7-8,10) ATTACKED BY WPNS OF TYPE (ALL    ) MUST BE EQ  1,

*       4       1.00        100.00         0.00          -8.00         -4    CONV DA
MAGE
```

```
  NO. OF TGTS OF TYPE (11-14) ATTACKED BY WPNS OF TYPE (ALL    ) MUST BE EQ  1,

*       5        0.70        69.57          0.00          -8.00          -5   IND DAM
AGE             *** NOT ACHIEVED
  NO. OF TGTS OF TYPE (15-17) ATTACKED BY WPNS OF TYPE (ALL    ) MUST BE EQ  1,

*       6        0.59        59.27          1.00          -8.00          -6   ALL TAR
GETS            *** NOT ACHIEVED
  VALUE DEST. ON TGTS (1-17) BY WPNS OF TYPE ( ALL    ) MUST BE EQ  1
```

```
(H-5)                                    SUPPLEMENTARY ACCOUNTING STATISTICS


   LEVEL     PERCENT
                                                                        CONDIT
ION ACHIEVED  ACHIEVED    TASK
  VALUE DEST. ON TGTS (1-3) BY WPNS OF TYPE ( ALL    ) MUST BE LE  1
 1     0.86     85.86     LC3I TRGTS
  VALUE DEST. ON TGTS (4-10) BY WPNS OF TYPE ( ALL    ) MUST BE LE  1
 2     0.43     43.45     NUC TRGTS
  VALUE DEST. ON TGTS (11-14) BY WPNS OF TYPE ( ALL    ) MUST BE LE  1
 3     0.75     75.04     CONV TRGTS
  VALUE DEST. ON TGTS (15-17) BY WPNS OF TYPE ( ALL    ) MUST BE LE  1
 4     0.51     50.67     IND/ECON TRG
  VALUE DEST. ON TGTS (1-17) BY WPNS OF TYPE ( ALL    ) MUST BE LE  1
 5     0.59     59.24     TOTAL TRGTS
```

```
BASE FILE = IAEM1                          OUTPUT FILE = OAEM1

FINISHED - CONDITION CODE =     0
```

Appendix L: The STORAGE.DAT File

The STORAGE.DAT file is created and "stored" with data by the Tool Two program (Appendix D). The file is "stored" two lines at a time (one AEM execution) where the first line lists the values for the AEM variables (weapons in this case) and the second line gives the values generated by the AEM for the dependent variables.

```
450 550  50 100 304 312 336  78  78  98 30 30  65  65  65  62
84.66 88.60 81.69 86.[illegible] 85.75
450 550  50 100 304 312   0  78  78  98 30 30  65  65  65  62
88.86 61.21 76.16 75.93 74.36
450 550  50 100 176 312   0  78  78  98 30 30  65  65  65  62
86.77 61.21 74.65 63.67 68.57
450 550  50 100 176 312 336  78  78  98 30 30  65  65  65  62
85.37 88.69 81.88 79.82 82.90
450 100  50 100 176 312 336  78  78  98 30 30  65  65  65  62
84.96 88.60 80.28 74.79 80.32
450 100  50 100 304 312 336  78  78  98 30 30  65  65  65  62
84.06 88.67 80.24 82.21 83.35
  0 100  50 100 304 312 336  78  78  98 30 30  65  65  65  62
84.66 88.60 79.13 80.78 82.53
  0 550  50 100 304 312 336  78  78  98 30 30  65  65  65  62
83.77 88.60 82.33 85.08 85.01
  0 550  50 100 304 312   0  78  78  98 30 30  65  65  65  62
89.08 61.21 75.54 73.55 73.23
  0 550  50 100 176 312   0  78  78  98 30 30  65  65  65  62
87.91 61.21 75.50 52.32 64.12
  0 550  50 100 176 312 336  78  78  98 30 30  65  65  65  62
83.81 88.65 82.33 78.28 82.16
  0 100  50 100 176 312 336  78  78  98 30 30  65  65  65  62
83.11 88.60 78.75 74.20 79.48
  0 100  50 100 176 312   0  78  78  98 30 30  65  65  65  62
82.82 42.99 76.01 45.35 56.77
  0 100  50 100 304 312   0  78  78  98 30 30  65  65  65  62
82.82 42.99 76.01 70.95 67.57
450 100  50 100 304 312   0  78  78  98 30 30  65  65  65  62
88.53 48.64 76.01 70.95 69.47
450 100  50 100 176 312   0  78  78  98 30 30  65  65  65  62
88.53 48.64 76.01 45.35 58.67
225 550 100 100 240 384 168 156 156  98 30 30  65  65  65  62
81.53 89.41 85.71 80.67 83.88
225 550 100 100 240 384 168   0   0  98 30 30  65  65  65  62
83.77 89.43 81.43 79.06 82.44
```

```
225 550 100 100 240 240 168   0   0  98 30 30  65  65  65  62
83.37 89.43 78.52 73.72 79.43
225 550 100 100 240 240 168 156 156  98 30 30  65  65  65  62
84.66 89.53 80.19 76.14 81.03
225 550   0 100 240 240 168 156 156  98 30 30  65  65  65  62
86.15 77.89 74.77 67.77 73.86
225 550   0 100 240 384 168 156 156  98 30 30  65  65  65  62
88.85 77.89 75.92 76.82 78.27
225 100   0 100 240 384 168 156 156  98 30 30  65  65  65  62
86.36 69.77 75.74 73.78 74.90
225 100 100 100 240 384 168 156 156  98 30 30  65  65  65  62
84.74 89.43 78.77 77.49 81.25
225 100 100 100 240 384 168   0   0  98 30 30  65  65  65  62
84.46 89.40 76.43 74.97 79.58
225 100 100 100 240 240 168   0   0  98 30 30  65  65  65  62
83.95 89.43 74.41 60.66 72.99
225 100 100 100 240 240 168 156 156  98 30 30  65  65  65  62
82.05 89.36 76.03 72.44 78.11
225 100   0 100 240 240 168 156 156  98 30 30  65  65  65  62
84.04 69.77 75.70 55.37 66.84
225 100   0 100 240 240 168   0   0  98 30 30  65  65  65  62
86.34 69.77 76.32 37.22 59.61
225 100   0 100 240 384 168   0   0  98 30 30  65  65  65  62
86.34 69.77 76.32 61.26 69.76
225 550   0 100 240 384 168   0   0  98 30 30  65  65  65  62
88.97 77.89 75.30 72.98 76.52
225 550   0 100 240 240 168   0   0  98 30 30  65  65  65  62
88.95 77.89 75.26 49.94 66.78
225 325 100 200 240 312 336  78  78 156 30 30  65  65  65  62
84.07 89.22 83.41 86.59 86.08
225 325 100 200 240 312 336  78  78  40 30 30  65  65  65  62
84.07 89.22 83.29 82.31 84.25
225 325 100 200 240 312   0  78  78  40 30 30  65  65  65  62
87.10 75.78 78.96 42.13 63.71
225 325 100 200 240 312   0  78  78 156 30 30  65  65  65  62
87.80 75.78 73.84 72.53 75.38
225 325 100   0 240 312   0  78  78 156 30 30  65  65  65  62
87.97 69.83 73.84 72.53 74.12
225 325 100   0 240 312 336  78  78 156 30 30  65  65  65  62
84.96 89.68 86.04 83.00 85.41
225 325   0   0 240 312 336  78  78 156 30 30  65  65  65  62
84.68 88.03 79.56 77.43 81.10
225 325   0 200 240 312 336  78  78 156 30 30  65  65  65  62
84.88 87.57 80.73 78.77 81.88
225 325   0 200 240 312 336  78  78  40 30 30  65  65  65  62
84.73 87.57 78.11 75.37 79.79
225 325   0 200 240 312   0  78  78  40 30 30  65  65  65  62
87.10 45.44 78.96 42.13 57.17
225 325   0 200 240 312   0  78  78 156 30 30  65  65  65  62
87.80 45.44 73.84 72.53 68.83
225 325   0   0 240 312   0  78  78 156 30 30  65  65  65  62
81.33 42.99 73.65 72.64 67.53
225 325   0   0 240 312   0  78  78  40 30 30  65  65  65  62
```

```
80.63 42.99 78.96 42.13 55.86
225 325   0   0 240 312 336  78  78  40 30 30  65  65  65  62
84.72 88.03 78.33 73.09 78.97
225 325 100   0 240 312 336  78  78  40 30 30  65  65  65  62
84.61 89.74 82.10 81.02 83.59
225 325 100   0 240 312   0  78  78  40 30 30  65  65  65  62
87.27 69.83 78.96 42.13 62.45
225 325  50 200 304 312 168 156 156  98 60 60  65  65  65  62
86.60 88.53 77.85 76.88 80.80
225 325  50 200 304 312 168 156 156  98  0  0  65  65  65  62
86.68 88.53 75.20 76.89 80.16
225 325  50 200 304 312 168   0   0  98  0  0  65  65  65  62
86.68 88.53 75.68 72.27 78.33
225 325  50 200 304 312 168   0   0  98 60 60  65  65  65  62
86.68 88.53 74.57 74.71 79.09
225 325  50 200 176 312 168   0   0  98 60 60  65  65  65  62
86.68 88.53 74.53 54.68 70.63
225 325  50 200 176 312 168 156 156  98 60 60  65  65  65  62
83.21 88.53 74.35 71.84 77.41
225 325  50   0 176 312 168 156 156  98 60 60  65  65  65  62
86.11 88.67 75.13 58.45 72.33
225 325  50   0 304 312 168 156 156  98 60 60  65  65  65  62
88.43 88.67 75.18 75.00 79.60
225 325  50   0 304 312 168 156 156  98  0  0  65  65  65  62
88.11 88.67 76.31 72.64 78.84
225 325  50   0 304 312 168   0   0  98  0  0  65  65  65  62
88.41 88.67 76.89 58.18 72.92
225 325  50   0 304 312 168   0   0  98 60 60  65  65  65  62
88.41 88.67 75.75 66.02 75.95
225 325  50   0 176 312 168   0   0  98 60 60  65  65  65  62
88.41 88.67 75.75 40.24 65.07
225 325  50   0 176 312 168   0   0  98  0  0  65  65  65  62
88.41 88.67 76.89 32.26 61.98
225 325  50   0 176 312 168 156 156  98  0  0  65  65  65  62
86.11 88.67 76.27 50.46 69.23
225 325  50 200 176 312 168 156 156  98  0  0  65  65  65  62
84.31 88.53 75.06 64.81 74.75
225 325  50 200 176 312 168   0   0  98  0  0  65  65  65  62
86.68 88.53 75.66 46.70 67.54
225 325  50 100 304 384 168  78  78 156 30 30 100 100 100  62
86.45 89.14 81.05 80.87 83.37
225 325  50 100 304 384 168  78  78 156 30 30  30  30  30  62
87.21 81.41 73.74 79.57 79.47
225 325  50 100 304 384 168  78  78  40 30 30  30  30  30  62
85.64 81.42 78.23 67.73 75.37
225 325  50 100 304 384 168  78  78  40 30 30 100 100 100  62
87.60 89.17 76.73 77.71 81.13
225 325  50 100 304 240 168  78  78  40 30 30 100 100 100  62
86.45 89.17 76.73 67.69 76.76
225 325  50 100 304 240 168  78  78 156 30 30 100 100 100  62
87.41 89.14 81.92 72.91 80.34
225 325  50 100 176 240 168  78  78 156 30 30 100 100 100  62
85.00 89.05 73.61 67.03 75.53
```

```
225 325  50 100 176 384 168  78  78 156 30 30 100 100 100  62
87.41 89.14 80.17 73.93 80.34
225 325  50 100 176 384 168  78  78 156 30 30  30  30  30  62
85.70 81.40 73.47 72.64 76.29
225 325  50 100 176 384 168  78  78  40 30 30  30  30  30  62
85.64 81.42 78.23 42.03 64.53
225 325  50 100 176 384 168  78  78  40 30 30 100 100 100  62
86.45 89.18 76.73 65.92 76.02
225 325  50 100 176 240 168  78  78  40 30 30 100 100 100  62
86.45 89.17 76.73 41.99 65.92
225 325  50 100 176 240 168  78  78  40 30 30  30  30  30  62
86.79 81.42 76.94 18.19 54.30
225 325  50 100 176 240 168  78  78 156 30 30  30  30  30  62
85.70 81.40 73.47 48.91 66.28
225 325  50 100 304 240 168  78  78 156 30 30  30  30  30  62
86.84 81.41 73.74 72.38 76.39
225 325  50 100 304 240 168  78  78  40 30 30  30  30  30  62
86.79 81.42 76.94 44.13 65.24
225 325  50 100 240 384 336  78  78  98 60 60  65  65  65 124
84.96 89.46 81.88 88.04 86.48
225 325  50 100 240 384 336  78  78  98 60 60  65  65  65   0
84.07 68.03 84.88 81.88 79.88
225 325  50 100 240 384 336  78  78  98  0  0  65  65  65   0
83.83 67.97 84.65 80.92 79.38
225 325  50 100 240 384 336  78  78  98  0  0  65  65  65 124
84.07 89.46 81.11 87.55 85.98
225 325  50 100 240 384   0  78  78  98  0  0  65  65  65 124
89.10 58.59 77.88 76.32 74.40
225 325  50 100 240 384   0  78  78  98 60 60  65  65  65 124
89.07 58.56 77.87 77.77 75.00
225 325  50 100 240 240   0  78  78  98 60 60  65  65  65 124
86.68 58.56 74.94 72.64 71.84
225 325  50 100 240 240 336  78  78  98 60 60  65  65  65 124
82.80 89.41 81.66 82.55 83.84
225 325  50 100 240 240 336  78  78  98 60 60  65  65  65   0
83.77 68.01 81.01 77.59 77.09
225 325  50 100 240 240 336  78  78  98  0  0  65  65  65   0
84.46 68.01 80.57 76.35 76.55
225 325  50 100 240 240 336  78  78  98  0  0  65  65  65 124
83.82 89.41 81.33 81.27 83.35
225 325  50 100 240 240   0  78  78  98  0  0  65  65  65 124
86.46 58.54 74.44 70.20 70.66
225 325  50 100 240 240   0  78  78  98  0  0  65  65  65   0
87.93 37.13 76.58 42.30 54.97
225 325  50 100 240 240   0  78  78  98 60 60  65  65  65   0
87.93 37.13 75.44 50.29 58.06
225 325  50 100 240 384   0  78  78  98 60 60  65  65  65   0
88.65 37.13 75.49 72.79 67.65
225 325  50 100 240 384   0  78  78  98  0  0  65  65  65   0
87.93 37.13 76.58 66.21 65.05
450 325  50 100 240 312 336 156 156  98 30 30 100 100 100  62
84.73 89.01 82.63 84.82 85.18
450 325  50 100 240 312 336 156 156  98 30 30  30  30  30  62
```

```
82.77 81.29 83.20 80.70 81.68
450 325  50 100 240 312 336   0   0  98 30 30  30  30  30  62
84.66 81.34 80.02 78.45 80.20
450 325  50 100 240 312 336   0   0  98 30 30 100 100 100  62
84.96 89.06 82.62 81.38 83.76
450 325  50 100 240 312   0   0   0  98 30 30 100 100 100  62
88.84 61.62 74.98 67.60 70.65
450 325  50 100 240 312   0 156 156  98 30 30 100 100 100  62
88.61 61.61 78.60 73.15 73.83
  0 325  50 100 240 312   0 156 156  98 30 30 100 100 100  62
88.72 53.62 78.60 72.95 72.04
  0 325  50 100 240 312 336 156 156  98 30 30 100 100 100  62
84.73 89.01 82.15 83.36 84.45
  0 325  50 100 240 312 336 156 156  98 30 30  30  30  30  62
83.77 81.24 81.40 79.71 80.93
  0 325  50 100 240 312 336   0   0  98 30 30  30  30  30  62
83.89 81.24 79.66 76.99 79.38
  0 325  50 100 240 312 336   0   0  98 30 30 100 100 100  62
84.79 89.01 82.75 79.63 83.03
  0 325  50 100 240 312   0   0   0  98 30 30 100 100 100  62
89.55 53.64 74.98 66.43 68.51
  0 325  50 100 240 312   0   0   0  98 30 30  30  30  30  62
89.31 45.87 76.32 43.03 57.26
  0 325  50 100 240 312   0 156 156  98 30 30  30  30  30  62
86.59 45.87 76.01 61.19 64.52
450 325  50 100 240 312   0 156 156  98 30 30  30  30  30  62
85.88 53.85 76.01 62.37 66.66
450 325  50 100 240 312   0   0   0  98 30 30  30  30  30  62
88.59 53.85 76.32 44.24 59.41
225 550  50 100 240 312 168 156 156 156 30 30  65  65  65 124
85.12 89.58 83.05 80.07 83.45
225 550  50 100 240 312 168 156 156 156 30 30  65  65  65   0
87.60 68.14 79.79 75.88 76.56
225 550  50 100 240 312 168 156 156  40 30 30  65  65  65   0
85.85 68.14 76.62 72.94 74.34
225 550  50 100 240 312 168 156 156  40 30 30  65  65  65 124
87.42 89.57 77.01 78.28 81.50
225 550  50 100 240 312 168   0   0  40 30 30  65  65  65 124
87.42 89.58 76.90 74.46 79.87
225 550  50 100 240 312 168   0   0 156 30 30  65  65  65 124
87.41 89.54 79.87 77.73 81.96
225 100  50 100 240 312 168   0   0 156 30 30  65  65  65 124
86.18 85.66 78.81 73.56 78.96
225 100  50 100 240 312 168 156 156 156 30 30  65  65  65 124
86.18 85.66 79.41 77.05 80.57
225 100  50 100 240 312 168 156 156 156 30 30  65  65  65   0
85.51 64.21 76.51 72.95 73.43
225 100  50 100 240 312 168 156 156  40 30 30  65  65  65   0
82.99 64.24 78.83 51.21 64.53
225 100  50 100 240 312 168 156 156  40 30 30  65  65  65 124
86.25 85.66 77.29 73.20 78.45
225 100  50 100 240 312 168   0   0  40 30 30  65  65  65 124
86.34 85.68 75.06 65.39 74.62
```

```
225 100  50 100 240 312 168   0   0  40 30 30  65  65  65   0
85.95 64.24 78.81 33.09 57.24
225 100  50 100 240 312 168   0   0 156 30 30  65  65  65   0
86.34 64.24 73.77 64.59 69.35
225 550  50 100 240 312 168   0   0 156 30 30  65  65  65   0
87.12 68.10 78.79 72.71 74.92
225 550  50 100 240 312 168   0   0  40 30 30  65  65  65   0
87.60 68.14 76.66 57.84 68.19
450 325 100 100 240 312 168  78  78 156 60 60  65  65  65  62
85.07 89.40 80.82 78.28 82.11
450 325 100 100 240 312 168  78  78 156  0  0  65  65  65  62
84.66 89.40 79.60 77.75 81.54
450 325 100 100 240 312 168  78  78  40  0  0  65  65  65  62
83.77 89.40 78.14 73.83 79.43
450 325 100 100 240 312 168  78  78  40 60 60  65  65  65  62
84.55 89.40 78.10 75.02 80.01
450 325   0 100 240 312 168  78  78  40 60 60  65  65  65  62
86.88 77.89 78.27 48.09 66.49
450 325   0 100 240 312 168  78  78 156 60 60  65  65  65  62
87.53 77.85 75.87 72.56 76.30
  0 325   0 100 240 312 168  78  78 156 60 60  65  65  65  62
86.59 74.02 75.15 72.55 75.18
  0 325 100 100 240 312 168  78  78 156 60 60  65  65  65  62
84.66 89.40 79.80 77.02 81.28
  0 325 100 100 240 312 168  78  78 156  0  0  65  65  65  62
84.66 89.40 79.84 75.59 80.69
  0 325 100 100 240 312 168  78  78  40  0  0  65  65  65  62
83.46 89.41 76.74 72.59 78.53
  0 325 100 100 240 312 168  78  78  40 60 60  65  65  65  62
84.33 89.43 78.17 72.97 79.14
  0 325   0 100 240 312 168  78  78  40 60 60  65  65  65  62
85.87 74.06 78.30 46.79 65.00
  0 325   0 100 240 312 168  78  78  40  0  0  65  65  65  62
85.87 74.06 76.55 41.72 62.43
  0 325   0 100 240 312 168  78  78 156  0  0  65  65  65  62
86.68 74.06 74.03 69.53 73.65
450 325   0 100 240 312 168  78  78 156  0  0  65  65  65  62
87.69 77.89 74.03 70.70 75.09
450 325   0 100 240 312 168  78  78  40  0  0  65  65  65  62
86.88 77.90 77.66 41.72 63.65
225 550  50 200 240 312 168  78  78  98 60 60 100 100 100  62
86.50 88.91 79.05 77.68 81.49
225 550  50 200 240 312 168  78  78  98 60 60  30  30  30  62
85.98 81.18 78.36 73.08 77.65
225 550  50 200 240 312 168  78  78  98  0  0  30  30  30  62
84.93 81.14 77.24 72.59 77.04
225 550  50 200 240 312 168  78  78  98  0  0 100 100 100  62
86.59 88.91 78.99 76.30 80.91
225 550  50   0 240 312 168  78  78  98  0  0 100 100 100  62
88.53 89.37 79.37 73.04 79.96
225 550  50   0 240 312 168  78  78  98 60 60 100 100 100  62
88.53 89.40 79.22 74.51 80.54
225 100  50   0 240 312 168  78  78  98 60 60 100 100 100  62
```

```
85.03 82.63 77.14 72.63 77.37
225 100  50 200 240 312 168  78  78  98 60 60 100 100 100  62
85.21 87.93 77.15 72.55 78.50
225 100  50 200 240 312 168  78  78  98 60 60  30  30  30  62
84.53 80.14 75.75 56.10 69.46
225 100  50 200 240 312 168  78  78  98  0  0  30  30  30  62
84.53 80.14 76,89 48.12 66.37
225 100  50 200 240 312 168  78  78  98  0  0 100 100 100  62
85.16 87.91 75.25 71.36 77.52
225 100  50   0 240 312 168  78  78  98  0  0 100 100 100  62
85.16 82.65 75.25 71.36 76.39
225 100  50   0 240 312 168  78  78  98  0  0  30  30  30  62
84.53 74.88 76.89 48.12 65.23
225 100  50   0 240 312 168  78  78  98 60 60  30  30  30  62
84.53 74.88 75.75 56.10 68.32
225 550  50   0 240 312 168  78  78  98 60 60  30  30  30  62
86.70 81.64 74.34 72.15 76.47
225 550  50   0 240 312 168  78  78  98  0  0  30  30  30  62
86.70 81.64 75.48 64.39 73.47
225 325 100 100 304 312 168  78  78  98 30 30 100 100 100 124
84.12 89.40 81.59 86.16 85.51
225 325 100 100 304 312 168  78  78  98 30 30 100 100 100   0
83.77 76.13 79.78 80.30 79.69
225 325 100 100 304 312 168  78  78  98 30 30  30  30  30   0
83.89 61.41 78.52 77.87 75.20
225 325 100 100 304 312 168  78  78  98 30 30  30  30  30 124
83.00 90.62 79.72 80.86 82.94
225 325 100 100 176 312 168  78  78  98 30 30  30  30  30 124
83.63 90.65 78.85 73.93 79.89
225 325 100 100 176 312 168  78  78  98 30 30 100 100 100 124
84.96 89.35 81.62 79.18 82.66
225 325   0 100 176 312 168  78  78  98 30 30 100 100 100 124
86.83 76.60 78.53 73.71 77.07
225 325   0 100 304 312 168  78  78  98 30 30 100 100 100 124
88.24 76.60 78.96 80.36 80.15
225 325   0 100 304 312 168  78  78  98 30 30 100 100 100   0
88.43 62.94 76.05 73.32 73.55
225 325   0 100 304 312 168  78  78  98 30 30  30  30  30   0
86.60 48.22 76.32 64.92 66.68
225 325   0 100 304 312 168  78  78  98 30 30  30  30  30 124
87.76 77.44 74.05 75.40 76.99
225 325   0 100 176 312 168  78  78  98 30 30  30  30  30 124
86.60 77.42 74.00 58.42 69.67
225 325   0 100 176 312 168  78  78  98 30 30  30  30  30   0
86.60 48.22 76.32 39.14 55.80
225 325   0 100 176 312 168  78  78  98 30 30 100 100 100   0
87.26 62.94 76.01 51.56 64.22
225 325 100 100 176 312 168  78  78  98 30 30 100 100 100   0
81.94 76.16 79.79 73.47 76.60
225 325 100 100 176 312 168  78  78  98 30 30  30  30  30   0
81.72 61.00 75.80 72.72 72.01
450 325  50 200 240 384 168  78  78  98 30 30  65  65  65 124
85.23 89.31 81.10 79.48 82.68
```

```
450 325  50 200 240 384 168  78  78  98 30 30  65  65  65   0
86.56 67.91 76.81 75.90 75.67
450 325  50 200 240 240 168  78  78  98 30 30  65  65  65   0
84.17 67.91 74.55 66.17 70.73
450 325  50 200 240 240 168  78  78  98 30 30  65  65  65 124
86.52 89.31 78.30 73.73 79.73
450 325  50   0 240 240 168  78  78  98 30 30  65  65  65 124
86.49 89.77 75.62 72.55 78.68
450 325  50   0 240 384 168  78  78  98 30 30  65  65  65 124
88.40 89.81 77.99 78.06 81.82
  0 325  50   0 240 384 168  78  78  98 30 30  65  65  65 124
87.67 87.32 77.70 77.18 80.75
  0 325  50 200 240 384 168  78  78  98 30 30  65  65  65 124
85.64 89.35 78.35 78.92 81.84
  0 325  50 200 240 384 168  78  78  98 30 30  65  65  65   0
86.79 67.91 75.38 74.07 74.58
  0 325  50 200 240 240 168  78  78  98 30 30  65  65  65   0
85.64 67.91 75.33 54.98 66.38
  0 325  50 200 240 240 168  78  78  98 30 30  65  65  65 124
84.25 89.31 77.50 72.47 78.74
  0 325  50   0 240 240 168  78  78  98 30 30  65  65  65 124
85.26 87.32 74.39 72.08 77.51
  0 325  50   0 240 240 168  78  78  98 30 30  65  65  65   0
86.68 65.90 76.01 46.29 62.57
  0 325  50   0 240 384 168  78  78  98 30 30  65  65  65   0
86.68 65.90 76.01 70.09 72.61
450 325  50   0 240 384 168  78  78  98 30 30  65  65  65   0
88.39 68.37 75.74 72.93 74.48
450 325  50   0 240 240 168  78  78  98 30 30  65  65  65   0
87.25 68.37 75.70 51.15 65.14
225 325  50 100 240 312 168  78  78  98 30 30  65  65  65  62
86.45 88.76 75.51 64.99 75.24
```

Appendix M: The RESPNSES.IN File

The RESPNSES.IN file is manually created from the STORAGE.DAT file shown in Appendix L. Starting with line one of the STORAGE.DAT file, every other line is deleted through the whole file. What remains are the lines containing the dependent variable values from each of the 193 AEM executions, i.e. the RESPNSES.IN file.

84.66	88.60	81.69	86.93	85.75
88.86	61.21	76.16	75.93	74.36
86.77	61.21	74.65	63.67	68.57
85.37	88.69	81.88	79.82	82.90
84.96	88.60	80.28	74.79	80.32
84.06	88.67	80.24	82.21	83.35
84.66	88.60	79.13	80.78	82.53
83.77	88.60	82.33	85.08	85.01
89.08	61.21	75.54	73.55	73.23
87.91	61.21	75.50	52.32	64.12
83.81	88.65	82.33	78.28	82.16
83.11	88.60	78.75	74.20	79.48
82.82	42.99	76.01	45.35	56.77
82.82	42.99	76.01	70.95	67.57
88.53	48.64	76.01	70.95	69.47
88.53	48.64	76.01	45.35	58.67
81.53	89.41	85.71	80.67	83.88
83.77	89.43	81.43	79.06	82.44
83.37	89.43	78.52	73.72	79.43
84.66	89.53	80.19	76.14	81.03
86.15	77.89	74.77	67.77	73.86
88.85	77.89	75.92	76.82	78.27
86.36	69.77	75.74	73.78	74.90
84.74	89.43	78.77	77.49	81.25
84.46	89.40	76.43	74.97	79.58
83.95	89.43	74.41	60.66	72.99
82.05	89.36	76.03	72.44	78.11
84.04	69.77	75.70	55.37	66.84
86.34	69.77	76.32	37.22	59.61
86.34	69.77	76.32	61.26	69.76
88.97	77.89	75.30	72.98	76.52
88.95	77.89	75.26	49.94	66.78
84.07	89.22	83.41	86.59	86.08
84.07	89.22	83.29	82.31	84.25
87.10	75.78	78.96	42.13	63.71
87.80	75.78	73.84	72.53	75.38

87.97	69.83	73.84	72.53	74.12
84.96	89.68	86.04	83.00	85.41
84.68	88.03	79.56	77.43	81.10
84.88	87.57	80.73	78.77	81.88
84.73	87.57	78.11	75.37	79.79
87.10	45.44	78.96	42.13	57.17
87.80	45.44	73.84	72.53	68.83
81.33	42.99	73.65	72.64	67.53
80.63	42.99	78.96	42.13	55.86
84.72	88.03	78.33	73.09	78.97
84.61	89.74	82.10	81.02	83.59
87.27	69.83	78.96	42.13	62.45
86.60	88.53	77.85	76.88	80.80
86.68	88.53	75.20	76.89	80.16
86.68	88.53	75.68	72.27	78.33
86.68	88.53	74.57	74.71	79.09
86.68	88.53	74.53	54.68	70.63
83.21	88.53	74.35	71.84	77.41
86.11	88.67	75.13	58.45	72.33
88.43	88.67	75.18	75.00	79.60
88.11	88.67	76.31	72.64	78.84
88.41	88.67	76.89	58.18	72.92
88.41	88.67	75.75	66.02	75.95
88.41	88.67	75.75	40.24	65.07
88.41	88.67	76.89	32.26	61.98
86.11	88.67	76.27	50.46	69.23
84.31	88.53	75.06	64.81	74.75
86.68	88.53	75.66	46.70	67.54
86.45	89.14	81.05	80.87	83.37
87.21	81.41	73.74	79.57	79.47
85.64	81.42	78.23	67.73	75.37
87.60	89.17	76.73	77.71	81.13
86.45	89.17	76.73	67.69	76.76
87.41	89.14	81.92	72.91	80.34
85.00	89.05	73.61	67.03	75.53
87.41	89.14	80.17	73.93	80.34
85.70	81.40	73.47	72.64	76.29
85.64	81.42	78.23	42.03	64.53
86.45	89.18	76.73	65.92	76.02
86.45	89.17	76.73	41.99	65.92
86.79	81.42	76.94	18.19	54.30
85.70	81.40	73.47	48.91	66.28
86.84	81.41	73.74	72.38	76.39
86.79	81.42	76.94	44.13	65.24
84.96	89.46	81.88	88.04	86.48
84.07	68.03	84.88	81.88	79.88
83.83	67.97	84.65	80.92	79.38
84.07	89.46	81.11	87.55	85.98
89.10	58.59	77.88	76.32	74.40
89.07	58.56	77.87	77.77	75.00
86.68	58.56	74.94	72.64	71.84
82.80	89.41	81.66	82.55	83.84
83.77	68.01	81.01	77.59	77.09

84.46	68.01	80.57	76.35	76.55
83.82	89.41	81.33	81.27	83.35
86.46	58.54	74.44	70.20	70.66
87.93	37.13	76.58	42.30	54.97
87.93	37.13	75.44	50.29	58.06
88.65	37.13	75.49	72.79	67.65
87.93	37.13	76.58	66.21	65.05
84.73	89.01	82.63	84.82	85.18
82.77	81.29	83.20	80.70	81.68
84.66	81.34	80.02	78.45	80.20
84.96	89.06	82.62	81.38	83.76
88.84	61.62	74.98	67.60	70.65
88.61	61.61	78.60	73.15	73.83
88.72	53.62	78.60	72.95	72.04
84.73	89.01	82.15	83.36	84.45
83.77	81.24	81.40	79.71	80.93
83.89	81.24	79.66	76.99	79.38
84.79	89.01	82.75	79.63	83.03
89.55	53.64	74.98	66.43	68.51
89.31	45.87	76.32	43.03	57.26
86.59	45.87	76.01	61.19	64.52
85.88	53.85	76.01	62.37	66.66
88.59	53.85	76.32	44.24	59.41
85.12	89.58	83.05	80.07	83.45
87.60	68.14	79.79	75.88	76.56
85.85	68.14	76.62	72.94	74.34
87.42	89.57	77.01	78.28	81.50
87.42	89.58	76.90	74.46	79.87
87.41	89.54	79.87	77.73	81.96
86.18	85.66	78.81	73.56	78.96
86.18	85.66	79.41	77.05	80.57
85.51	64.21	76.51	72.95	73.43
82.99	64.24	78.83	51.21	64.53
86.25	85.66	77.29	73.20	78.45
86.34	85.68	75.06	65.39	74.62
85.95	64.24	78.81	33.09	57.24
86.34	64.24	73.77	64.59	69.35
87.12	68.10	78.79	72.71	74.92
87.60	68.14	76.66	57.84	68.19
85.07	89.40	80.82	78.28	82.11
84.66	89.40	79.60	77.75	81.54
83.77	89.40	78.14	73.83	79.43
84.55	89.40	78.10	75.02	80.01
86.88	77.89	78.27	48.09	66.49
87.53	77.85	75.87	72.56	76.30
86.59	74.02	75.15	72.55	75.18
84.66	89.40	79.80	77.02	81.28
84.66	89.40	79.84	75.59	80.69
83.46	89.41	76.74	72.59	78.53
84.33	89.43	78.17	72.97	79.14
85.87	74.06	78.30	46.79	65.00
85.87	74.06	76.55	41.72	62.43
86.68	74.06	74.03	69.53	73.65

87.69	77.89	74.03	70.70	75.09
86.88	77.90	77.66	41.72	63.65
86.50	88.91	79.05	77.68	81.49
85.98	81.18	78.36	73.08	77.65
84.93	81.14	77.24	72.59	77.04
86.59	88.91	78.99	76.30	80.91
88.53	89.37	79.37	73.04	79.96
88.53	89.40	79.22	74.51	80.54
85.03	82.63	77.14	72.63	77.37
85.21	87.93	77.15	72.55	78.50
84.53	80.14	75.75	56.10	69.46
84.53	80.14	76.89	48.12	66.37
85.16	87.91	75.25	71.36	77.52
85.16	82.65	75.25	71.36	76.39
84.53	74.88	76.89	48.12	65.23
84.53	74.88	75.75	56.10	68.32
86.70	81.64	74.34	72.15	76.47
86.70	81.64	75.48	64.39	73.47
84.12	89.40	81.59	86.16	85.51
83.77	76.13	79.78	80.30	79.69
83.89	61.41	78.52	77.87	75.20
83.00	90.62	79.72	80.86	82.94
83.63	90.65	78.85	73.93	79.89
84.96	89.35	81.62	79.18	82.66
86.83	76.60	78.53	73.71	77.07
88.24	76.60	78.96	80.36	80.15
88.43	62.94	76.05	73.32	73.55
86.60	48.22	76.32	64.92	66.68
87.76	77.44	74.05	75.40	76.99
86.60	77.42	74.00	58.42	69.67
86.60	48.22	76.32	39.14	55.80
87.26	62.94	76.01	51.56	64.22
81.94	76.16	79.79	73.47	76.60
81.72	61.00	75.80	72.72	72.01
85.23	89.31	81.10	79.48	82.68
86.56	67.91	76.81	75.90	75.67
84.17	67.91	74.55	66.17	70.73
86.52	89.31	78.30	73.73	79.73
86.49	89.77	75.62	72.55	78.68
88.40	89.81	77.99	78.06	81.82
87.67	87.32	77.70	77.18	80.75
85.64	89.35	78.35	78.92	81.84
86.79	67.91	75.38	74.07	74.58
85.64	67.91	75.33	54.98	66.38
84.25	89.31	77.50	72.47	78.74
85.26	87.32	74.39	72.08	77.51
86.68	65.90	76.01	46.29	62.57
86.68	65.90	76.01	70.09	72.61
88.39	68.37	75.74	72.93	74.48
87.25	68.37	75.70	51.15	65.14
86.45	88.76	75.51	64.99	75.24

Appendix N: The DSGNMTRX.IN File

The DSGNMTRX.IN file is created from the CODEDSGN subroutine when selection C is picked from the main menu of the RSMed-AEM program. The DSGNMTRX.IN file is the coded version (refer to equation 2.3) of the AEM.DAT file (Appendix J) with the exception of the first line. The first line first shows how many design variables were processed in the INPTDATA subroutine and then how many AEM variables there are in the IAEM1.DAT file (Appendix E). The expanded Box-Behnken experimental design follows.

```
12 16
  1.0  1.0  0.0  0.0  1.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0
  1.0  1.0  0.0  0.0  1.0  0.0 -1.0  0.0  0.0  0.0  0.0  0.0
  1.0  1.0  0.0  0.0 -1.0  0.0 -1.0  0.0  0.0  0.0  0.0  0.0
  1.0  1.0  0.0  0.0 -1.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0
  1.0 -1.0  0.0  0.0 -1.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0
  1.0 -1.0  0.0  0.0  1.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0
 -1.0 -1.0  0.0  0.0  1.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0
 -1.0  1.0  0.0  0.0  1.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0
 -1.0  1.0  0.0  0.0  1.0  0.0 -1.0  0.0  0.0  0.0  0.0  0.0
 -1.0  1.0  0.0  0.0 -1.0  0.0 -1.0  0.0  0.0  0.0  0.0  0.0
 -1.0  1.0  0.0  0.0 -1.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0
 -1.0 -1.0  0.0  0.0 -1.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0
 -1.0 -1.0  0.0  0.0 -1.0  0.0 -1.0  0.0  0.0  0.0  0.0  0.0
 -1.0 -1.0  0.0  0.0  1.0  0.0 -1.0  0.0  0.0  0.0  0.0  0.0
  1.0 -1.0  0.0  0.0  1.0  0.0 -1.0  0.0  0.0  0.0  0.0  0.0
  1.0 -1.0  0.0  0.0 -1.0  0.0 -1.0  0.0  0.0  0.0  0.0  0.0
  0.0  1.0  1.0  0.0  0.0  1.0  0.0  1.0  0.0  0.0  0.0  0.0
  0.0  1.0  1.0  0.0  0.0  1.0  0.0 -1.0  0.0  0.0  0.0  0.0
  0.0  1.0  1.0  0.0  0.0 -1.0  0.0 -1.0  0.0  0.0  0.0  0.0
  0.0  1.0  1.0  0.0  0.0 -1.0  0.0  1.0  0.0  0.0  0.0  0.0
  0.0  1.0 -1.0  0.0  0.0 -1.0  0.0  1.0  0.0  0.0  0.0  0.0
  0.0  1.0 -1.0  0.0  0.0  1.0  0.0  1.0  0.0  0.0  0.0  0.0
  0.0 -1.0 -1.0  0.0  0.0  1.0  0.0  1.0  0.0  0.0  0.0  0.0
  0.0 -1.0  1.0  0.0  0.0  1.0  0.0  1.0  0.0  0.0  0.0  0.0
  0.0 -1.0  1.0  0.0  0.0  1.0  0.0 -1.0  0.0  0.0  0.0  0.0
  0.0 -1.0  1.0  0.0  0.0 -1.0  0.0 -1.0  0.0  0.0  0.0  0.0
  0.0 -1.0  1.0  0.0  0.0 -1.0  0.0  1.0  0.0  0.0  0.0  0.0
  0.0 -1.0 -1.0  0.0  0.0 -1.0  0.0  1.0  0.0  0.0  0.0  0.0
  0.0 -1.0 -1.0  0.0  0.0 -1.0  0.0 -1.0  0.0  0.0  0.0  0.0
```

0.0	-1.0	-1.0	0.0	0.0	1.0	0.0	-1.0	0.0	0.0	0.0	0.0
0.0	1.0	-1.0	0.0	0.0	1.0	0.0	-1.0	0.0	0.0	0.0	0.0
0.0	1.0	-1.0	0.0	0.0	-1.0	0.0	-1.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	-1.0	0.0	0.0	0.0
0.0	0.0	1.0	1.0	0.0	0.0	-1.0	0.0	-1.0	0.0	0.0	0.0
0.0	0.0	1.0	1.0	0.0	0.0	-1.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	1.0	-1.0	0.0	0.0	-1.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	1.0	-1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	-1.0	-1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	-1.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	-1.0	1.0	0.0	0.0	1.0	0.0	-1.0	0.0	0.0	0.0
0.0	0.0	-1.0	1.0	0.0	0.0	-1.0	0.0	-1.0	0.0	0.0	0.0
0.0	0.0	-1.0	1.0	0.0	0.0	-1.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	-1.0	-1.0	0.0	0.0	-1.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	-1.0	-1.0	0.0	0.0	-1.0	0.0	-1.0	0.0	0.0	0.0
0.0	0.0	-1.0	-1.0	0.0	0.0	1.0	0.0	-1.0	0.0	0.0	0.0
0.0	0.0	1.0	-1.0	0.0	0.0	1.0	0.0	-1.0	0.0	0.0	0.0
0.0	0.0	1.0	-1.0	0.0	0.0	-1.0	0.0	-1.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	-1.0	0.0	0.0
0.0	0.0	0.0	1.0	1.0	0.0	0.0	-1.0	0.0	-1.0	0.0	0.0
0.0	0.0	0.0	1.0	1.0	0.0	0.0	-1.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	1.0	-1.0	0.0	0.0	-1.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	1.0	-1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	-1.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	-1.0	1.0	0.0	0.0	1.0	0.0	-1.0	0.0	0.0
0.0	0.0	0.0	-1.0	1.0	0.0	0.0	-1.0	0.0	-1.0	0.0	0.0
0.0	0.0	0.0	-1.0	1.0	0.0	0.0	-1.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	-1.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	-1.0	0.0	-1.0	0.0	0.0
0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	1.0	0.0	-1.0	0.0	0.0
0.0	0.0	0.0	1.0	-1.0	0.0	0.0	1.0	0.0	-1.0	0.0	0.0
0.0	0.0	0.0	1.0	-1.0	0.0	0.0	-1.0	0.0	-1.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0
0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	-1.0	0.0
0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	-1.0	0.0	-1.0	0.0
0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	-1.0	0.0	1.0	0.0
0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0	-1.0	0.0	1.0	0.0
0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0	1.0	0.0	1.0	0.0
0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	1.0	0.0	1.0	0.0
0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0
0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	1.0	0.0	-1.0	0.0
0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	-1.0	0.0	-1.0	0.0
0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	-1.0	0.0	1.0	0.0
0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	-1.0	0.0	1.0	0.0
0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	-1.0	0.0	-1.0	0.0
0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	1.0	0.0	-1.0	0.0
0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0	1.0	0.0	-1.0	0.0
0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0	-1.0	0.0	-1.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	1.0
0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	-1.0

0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	-1.0	0.0	-1.0
0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	-1.0	0.0	1.0
0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0	-1.0	0.0	1.0
0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0	1.0	0.0	1.0
0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	1.0	0.0	1.0
0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	1.0	0.0	1.0
0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	1.0	0.0	-1.0
0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	-1.0	0.0	-1.0
0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	-1.0	0.0	1.0
0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	-1.0	0.0	1.0
0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	-1.0	0.0	-1.0
0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	1.0	0.0	-1.0
0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0	1.0	0.0	-1.0
0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0	-1.0	0.0	-1.0
1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	-1.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0	-1.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0	1.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	1.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	1.0	0.0
-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	1.0	0.0
-1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0
-1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	-1.0	0.0
-1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0	-1.0	0.0
-1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0	1.0	0.0
-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	1.0	0.0
-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	-1.0	0.0
-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	-1.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	-1.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	-1.0	0.0
0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0
0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	-1.0
0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0	-1.0
0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0	1.0
0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	1.0
0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	1.0
0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	1.0
0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0
0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	-1.0
0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0	-1.0
0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0	1.0
0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	1.0
0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	-1.0
0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	-1.0
0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	-1.0
0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	-1.0
1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0
1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0
1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0
1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0
1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0
1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0
-1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0

-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0
-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0
-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0
-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0
-1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0
-1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0
-1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0
1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0
1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0
0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0
0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0
0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0
0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0
0.0	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0
0.0	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0
0.0	-1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0
0.0	-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0
0.0	-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0
0.0	-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0
0.0	-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0
0.0	-1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0
0.0	-1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0
0.0	-1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0
0.0	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0	0.0
0.0	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0
0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0
0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0
0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0
0.0	0.0	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0
0.0	0.0	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
0.0	0.0	-1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
0.0	0.0	-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
0.0	0.0	-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0
0.0	0.0	-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0
0.0	0.0	-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0
0.0	0.0	-1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0
0.0	0.0	-1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0
0.0	0.0	-1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0
0.0	0.0	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0	-1.0
0.0	0.0	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0
1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0
1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0
1.0	0.0	0.0	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0
1.0	0.0	0.0	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0
1.0	0.0	0.0	-1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0
1.0	0.0	0.0	-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0
-1.0	0.0	0.0	-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0
-1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0
-1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0
-1.0	0.0	0.0	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0
-1.0	0.0	0.0	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0
-1.0	0.0	0.0	-1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	1.0

-1.0	0.0	0.0	-1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0
-1.0	0.0	0.0	-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0
1.0	0.0	0.0	-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0
1.0	0.0	0.0	-1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Appendix O: The REGRESS.DAT File

The REGRESS.DAT file is created by the RGRSSION subroutine. The file contains the SSE, SSR, SSTO, R^2, and residual statistics along with the parameter sum-of-squares. The parameter sum-of-squares are shown in ascending order by the parameter sum-of-squares as an aid to the user when choosing variables to include in the model that will be used for informational analysis. This appendix shows the regression results for five dependent variables.

Regression Output for Dependent Variable: Y1

Model Sum of Squares (SSR)	Error Sum of Squares (SSE)	Total Sum of Squares (SSTO)	R-Square
452.22	197.90	650.12	0.6956

Parameters Of Y1 Sorted In Ascending Order
By Their Sum Of Squares

Number	Variable		Type I Sum of Squares	Percent Of SSR	Summed Percent Of SSR
1	PKEEP	* B-52CMC	0.001	0.00	0.00
2	MICBM	* B-52CMC	0.002	0.00	0.00
3	MMII	* B-52CMC	0.002	0.00	0.00
4	FB111PEN	* ATB	0.003	0.00	0.00
5	MMIII	* B-52PEN	0.004	0.00	0.00
6	MMII	* FB111PEN	0.004	0.00	0.00
7	MICBM	* FB111PEN	0.004	0.00	0.00
8	MICBM	* ATB	0.007	0.00	0.01
9	B1B	* ATB	0.020	0.00	0.01
10	POSC3	* B1B	0.021	0.00	0.02
11	POSTRIC4	* POSTRIC4	0.026	0.01	0.02
12	TRID5	* ATB	0.026	0.01	0.03
13	PKEEP	* B1B	0.026	0.01	0.03
14	MMII	* POSTRIC4	0.040	0.01	0.04
15	B-52PEN	* FB111PEN	0.046	0.01	0.05

Parameters Of Y1 Sorted In Ascending Order
By Their Sum Of Squares

Number	Variable		Type I Sum of Squares	Percent Of SSR	Summed Percent Of SSR
16	MMII	* MMII	0.047	0.01	0.06
17	MICBM	* MICBM	0.055	0.01	0.07
18	ATB	* ATB	0.061	0.01	0.09
19	MMII	* B1B	0.064	0.01	0.10
20	MICBM	* B-52PEN	0.068	0.01	0.12
21	MMIII	* FB111PEN	0.068	0.01	0.13
22	FB111PEN		0.073	0.02	0.15
23	PKEEP	* POSC3	0.092	0.02	0.17
24	POSC3	* POSC3	0.092	0.02	0.19
25	FB111PEN	* B1B	0.093	0.02	0.21
26	TRID5	* B1B	0.095	0.02	0.23
27	B-52CMC	* B1B	0.101	0.02	0.25
28	MICBM	* B1B	0.106	0.02	0.28
29	MMII	* B-52PEN	0.111	0.02	0.30
30	POSC3	* FB111PEN	0.112	0.02	0.32
31	MICBM	* POSC3	0.130	0.03	0.35
32	TRID5	* FB111PEN	0.139	0.03	0.38
33	B-52CMC	* FB111PEN	0.139	0.03	0.41
34	B1B	* B1B	0.150	0.03	0.45
35	FB111PEN	* FB111PEN	0.196	0.04	0.49
36	MICBM	* POSTRIC4	0.207	0.05	0.54
37	POSC3	* POSTRIC4	0.214	0.05	0.58
38	MMIII	* POSC3	0.216	0.05	0.63
39	ATB		0.248	0.05	0.69
40	B-52PEN	* B-52CMC	0.292	0.06	0.75
41	MMII	* POSC3	0.303	0.07	0.82
42	PKEEP	* FB111PEN	0.334	0.07	0.89
43	TRID5	* B-52CMC	0.342	0.08	0.97
44	PKEEP	* B-52PEN	0.432	0.10	1.06
45	B-52PEN	* ATB	0.449	0.10	1.16
46	MMII	* TRID5	0.543	0.12	1.28
47	POSC3	* B-52CMC	0.544	0.12	1.40
48	POSTRIC4	* B-52CMC	0.551	0.12	1.52
49	MMII	* PKEEP	0.574	0.13	1.65
50	MMII	* ATB	0.656	0.15	1.80
51	POSTRIC4	* FB111PEN	0.685	0.15	1.95
52	POSC3	* TRID5	0.706	0.16	2.10
53	MMIII	* B1B	0.722	0.16	2.26
54	POSTRIC4	* B-52PEN	0.833	0.18	2.45
55	POSTRIC4	* TRID5	0.842	0.19	2.63
56	MMIII	* B-52CMC	0.865	0.19	2.83
57	PKEEP	* ATB	0.926	0.20	3.03
58	POSC3	* ATB	1.035	0.23	3.26
59	B-52CMC	* B-52CMC	1.040	0.23	3.49
60	MMII	* MICBM	1.040	0.23	3.72
61	TRID5	* B-52PEN	1.097	0.24	3.96

Parameters Of Y1 Sorted In Ascending Order By Their Sum Of Squares

Number	Variable		Type I Sum of Squares	Percent Of SSR	Summed Percent Of SSR
62	POSTRIC4	* B1B	1.283	0.28	4.25
63	PKEEP	* POSTRIC4	1.305	0.29	4.53
64	POSTRIC4	* ATB	1.441	0.32	4.85
65	MMIII	* ATB	1.538	0.34	5.19
66	B-52CMC		1.802	0.40	5.59
67	MMIII	* POSTRIC4	1.911	0.42	6.01
68	B-52PEN	* B1B	2.318	0.51	6.53
69	MMIII	* MMIII	2.679	0.59	7.12
70	MMIII	* MICBM	2.756	0.61	7.73
71	B-52CMC	* ATB	2.822	0.62	8.35
72	B-52PEN	* B-52PEN	4.674	1.03	9.39
73	MMIII	* TRID5	5.176	1.14	10.53
74	POSC3	* B-52PEN	6.350	1.40	11.93
75	TRID5	* TRID5	6.466	1.43	13.36
76	MMII		6.734	1.49	14.85
77	MICBM		7.426	1.64	16.50
78	POSTRIC4		8.345	1.85	18.34
79	MMII	* MMIII	8.381	1.85	20.19
80	MMIII	* PKEEP	8.570	1.90	22.09
81	POSC3		9.579	2.12	24.21
82	B1B		10.898	2.41	26.62
83	MICBM	* TRID5	11.937	2.64	29.26
84	PKEEP	* TRID5	13.286	2.94	32.20
85	PKEEP	* MICBM	13.913	3.08	35.27
86	B-52PEN		18.922	4.18	39.46
87	MMIII		30.650	6.78	46.23
88	PKEEP	* PKEEP	34.983	7.74	53.97
89	PKEEP		67.261	14.87	68.84
90	TRID5		140.897	31.16	100.00

For Y1: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
1	84.66	83.87	0.79
2	88.86	88.66	0.20
3	86.77	87.51	-0.74
4	85.37	83.56	1.81
5	84.96	85.00	-0.04
6	84.06	84.84	-0.78
7	84.66	83.28	1.38
8	83.77	85.21	-1.44

For Y1: Residuals Associated With The Design Points

Design Point	ABM's Response	Postulated Model's Response	Residual
9	89.08	89.47	-0.39
10	87.91	87.77	0.14
11	83.81	84.35	-0.54
12	83.11	82.88	0.23
13	82.82	84.03	-1.21
14	82.82	85.27	-2.45
15	88.53	87.35	1.18
16	88.53	86.67	1.86
17	81.53	83.81	-2.28
18	83.77	84.09	-0.32
19	83.37	85.08	-1.71
20	84.66	83.89	0.77
21	86.15	86.51	-0.36
22	88.85	87.56	1.29
23	86.36	85.39	0.97
24	84.74	84.56	0.18
25	84.46	84.88	-0.42
26	83.95	84.50	-0.55
27	82.05	83.26	-1.21
28	84.04	82.94	1.10
29	86.34	84.84	1.50
30	86.34	86.37	-0.03
31	88.97	88.50	0.47
32	88.95	88.36	0.59
33	84.07	79.69	4.38
34	84.07	79.66	4.41
35	87.10	85.88	1.22
36	87.80	86.50	1.30
37	87.97	87.34	0.63
38	84.96	83.99	0.97
39	84.68	85.98	-1.30
40	84.88	85.42	-0.54
41	84.73	85.41	-0.68
42	87.10	87.99	-0.89
43	87.80	88.58	-0.78
44	81.33	85.69	-4.36
45	80.63	85.06	-4.43
46	84.72	85.93	-1.21
47	84.61	83.91	0.70
48	87.27	86.68	0.59
49	86.60	87.08	-0.48
50	86.68	86.98	-0.30
51	86.68	86.83	-0.15
52	86.68	87.15	-0.47
53	86.68	87.28	-0.60
54	83.21	84.70	-1.49
55	86.11	85.71	0.40
56	88.43	87.73	0.70

For Y1: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
57	88.11	87.58	0.53
58	88.41	87.17	1.24
59	88.41	87.54	0.87
60	88.41	88.04	0.37
61	88.41	88.01	0.40
62	86.11	85.90	0.21
63	84.31	84.93	-0.62
64	86.68	87.30	-0.62
65	86.45	88.28	-1.83
66	87.21	87.00	0.21
67	85.64	85.76	-0.12
68	87.60	87.37	0.23
69	86.45	86.68	-0.23
70	87.41	86.85	0.56
71	85.00	85.43	-0.43
72	87.41	87.32	0.09
73	85.70	86.14	-0.44
74	85.64	85.64	0.00
75	86.45	87.14	-0.69
76	86.45	85.99	0.46
77	86.79	85.63	1.16
78	85.70	85.38	0.32
79	86.84	86.70	0.14
80	86.79	86.21	0.58
81	84.96	85.07	-0.11
82	84.07	84.46	-0.39
83	83.83	84.14	-0.31
84	84.07	84.80	-0.73
85	89.10	87.96	1.14
86	89.07	88.60	0.47
87	86.68	86.58	0.10
88	82.80	83.97	-1.17
89	83.77	84.21	-0.44
90	84.46	84.72	-0.26
91	83.82	84.53	-0.71
92	86.46	86.77	-0.31
93	87.93	87.12	0.81
94	87.93	86.99	0.94
95	88.65	88.16	0.49
96	87.93	87.46	0.47
97	84.73	85.40	-0.67
98	82.77	83.84	-1.07
99	84.66	85.33	-0.67
100	84.96	85.37	-0.41
101	88.84	89.27	-0.43
102	88.61	88.26	0.35
103	88.72	87.39	1.33
104	84.73	85.05	-0.32

For Y1: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
105	83.77	83.74	0.03
106	83.89	84.90	-1.01
107	84.79	84.69	0.10
108	89.55	88.07	1.48
109	89.31	87.98	1.33
110	86.59	85.77	0.82
111	85.88	86.39	-0.51
112	88.59	88.93	-0.34
113	85.12	86.06	-0.94
114	87.60	87.06	0.54
115	85.85	86.08	-0.23
116	87.42	86.76	0.66
117	87.42	87.76	-0.34
118	87.41	86.52	0.89
119	86.18	86.24	-0.06
120	86.18	85.74	0.44
121	85.51	85.50	0.01
122	82.99	83.59	-0.60
123	86.25	85.51	0.74
124	86.34	86.55	-0.21
125	85.95	85.30	0.65
126	86.34	86.67	-0.33
127	87.12	88.19	-1.07
128	87.60	87.75	-0.15
129	85.07	84.94	0.13
130	84.66	84.74	-0.08
131	83.77	84.23	-0.46
132	84.55	84.81	-0.26
133	86.88	86.96	-0.08
134	87.53	87.07	0.46
135	86.59	86.04	0.55
136	84.66	84.67	-0.01
137	84.66	84.53	0.13
138	83.46	83.97	-0.51
139	84.33	84.48	-0.15
140	85.87	85.88	-0.01
141	85.87	85.94	-0.07
142	86.68	86.48	0.20
143	87.69	87.45	0.24
144	86.88	86.96	-0.08
145	86.50	86.31	0.19
146	85.98	85.38	0.60
147	84.93	85.05	-0.12
148	86.59	86.29	0.30
149	88.53	87.94	0.59
150	88.53	88.01	0.52
151	85.03	85.24	-0.21
152	85.21	85.20	0.01

For Y1: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
153	84.53	85.12	-0.59
154	84.53	85.05	-0.52
155	85.16	85.44	-0.28
156	85.16	85.43	-0.27
157	84.53	84.72	-0.19
158	84.53	84.83	-0.30
159	86.70	86.75	-0.05
160	86.70	86.38	0.32
161	84.12	85.10	-0.98
162	83.77	84.93	-1.16
163	83.89	84.21	-0.32
164	83.00	84.24	-1.24
165	83.63	84.17	-0.54
166	84.96	84.94	0.02
167	86.83	86.44	0.39
168	88.24	86.90	1.34
169	88.43	87.70	0.73
170	86.60	86.81	-0.21
171	87.76	85.87	1.89
172	86.60	85.51	1.09
173	86.60	85.43	1.17
174	87.26	86.21	1.05
175	81.94	83.75	-1.81
176	81.72	83.13	-1.41
177	85.23	86.73	-1.50
178	86.56	85.82	0.74
179	84.17	85.85	-1.68
180	86.52	85.91	0.61
181	86.49	86.92	-0.43
182	88.40	88.19	0.21
183	87.67	86.73	0.94
184	85.64	86.29	-0.65
185	86.79	86.19	0.60
186	85.64	86.02	-0.38
187	84.25	85.27	-1.02
188	85.26	85.26	0.00
189	86.68	85.92	0.76
190	86.68	86.54	0.14
191	88.39	87.20	1.19
192	87.25	86.77	0.48
193	86.45	86.45	0.00

Regression Output for Dependent Variable: Y2

Model Sum of Squares (SSR)	Error Sum of Squares (SSE)	Total Sum of Squares (SSTO)	R-Square
39535.10	181.91	39717.00	0.9954

Parameters Of Y2 Sorted In Ascending Order
By Their Sum Of Squares

Number	Variable		Type I Sum of Squares	Percent Of SSR	Summed Percent Of SSR
1	MICBM	* POSC3	0.000	0.00	0.00
2	MICBM	* POSTRIC4	0.000	0.00	0.00
3	MICBM	* B-52PEN	0.000	0.00	0.00
4	POSC3	* B-52PEN	0.000	0.00	0.00
5	POSC3	* FB111PEN	0.000	0.00	0.00
6	B-52PEN	* FB111PEN	0.000	0.00	0.00
7	MMII	* POSTRIC4	0.000	0.00	0.00
8	B-52PEN	* ATB	0.000	0.00	0.00
9	PKEEP	* B-52CMC	0.000	0.00	0.00
10	POSTRIC4	* FB111PEN	0.000	0.00	0.00
11	FB111PEN	* B1B	0.000	0.00	0.00
12	TRID5	* ATB	0.000	0.00	0.00
13	POSTRIC4	* B-52PEN	0.000	0.00	0.00
14	MMIII	* ATB	0.000	0.00	0.00
15	B-52CMC	* ATB	0.000	0.00	0.00
16	MMIII	* B-52CMC	0.000	0.00	0.00
17	FB111PEN		0.000	0.00	0.00
18	MMII	* FB111PEN	0.000	0.00	0.00
19	MMII	* B-52CMC	0.000	0.00	0.00
20	B-52PEN	* B1B	0.000	0.00	0.00
21	MMII	* POSC3	0.000	0.00	0.00
22	POSTRIC4	* TRID5	0.000	0.00	0.00
23	B-52PEN		0.000	0.00	0.00
24	MICBM	* FB111PEN	0.000	0.00	0.00
25	MMII	* ATB	0.000	0.00	0.00
26	PKEEP	* B-52PEN	0.000	0.00	0.00
27	MICBM	* ATB	0.000	0.00	0.00
28	MICBM	* B1B	0.000	0.00	0.00
29	MICBM	* B-52CMC	0.000	0.00	0.00
30	TRID5	* B-52CMC	0.000	0.00	0.00
31	FB111PEN	* ATB	0.000	0.00	0.00
32	MMIII	* FB111PEN	0.000	0.00	0.00
33	TRID5	* B-52PEN	0.000	0.00	0.00
34	POSC3	* TRID5	0.000	0.00	0.00
35	TRID5	* FB111PEN	0.000	0.00	0.00
36	TRID5	* B1B	0.000	0.00	0.00

Parameters Of Y2 Sorted In Ascending Order
By Their Sum Of Squares

Number	Variable		Type I Sum of Squares	Percent Of SSR	Summed Percent Of SSR
37	PKEEP	* POSTRIC4	0.000	0.00	0.00
38	POSTRIC4	* B-52CMC	0.000	0.00	0.00
39	B-52PEN	* B-52CMC	0.000	0.00	0.00
40	MMII	* B1B	0.001	0.00	0.00
41	MMIII	* B1B	0.001	0.00	0.00
42	B-52CMC	* FB111PEN	0.001	0.00	0.00
43	MMII	* B-52PEN	0.001	0.00	0.00
44	POSTRIC4	* B1B	0.001	0.00	0.00
45	POSC3	* POSTRIC4	0.001	0.00	0.00
46	PKEEP	* FB111PEN	0.001	0.00	0.00
47	POSTRIC4		0.001	0.00	0.00
48	POSC3	* B-52CMC	0.001	0.00	0.00
49	B-52CMC	* B1B	0.002	0.00	0.00
50	MMIII	* POSTRIC4	0.002	0.00	0.00
51	MMIII	* B-52PEN	0.002	0.00	0.00
52	POSTRIC4	* ATB	0.002	0.00	0.00
53	MMIII	* POSC3	0.003	0.00	0.00
54	POSC3		0.003	0.00	0.00
55	POSC3	* B1B	0.003	0.00	0.00
56	B-52CMC		0.005	0.00	0.00
57	POSC3	* ATB	0.007	0.00	0.00
58	PKEEP	* POSC3	0.009	0.00	0.00
59	PKEEP	* ATB	0.011	0.00	0.00
60	PKEEP	* B1B	0.011	0.00	0.00
61	PKEEP	* MICBM	3.010	0.01	0.01
62	PKEEP	* PKEEP	5.703	0.01	0.02
63	MMII	* MICBM	6.150	0.02	0.04
64	MMII	* MMIII	8.023	0.02	0.06
65	MMII	* PKEEP	14.765	0.04	0.10
66	MICBM	* TRID5	21.856	0.06	0.15
67	MICBM		23.900	0.06	0.21
68	MMIII	* MICBM	33.034	0.08	0.29
69	MMIII	* MMIII	47.537	0.12	0.41
70	MMII	* TRID5	57.566	0.15	0.56
71	MMIII	* PKEEP	65.206	0.16	0.73
72	MMII		73.681	0.19	0.91
73	POSTRIC4	* POSTRIC4	121.092	0.31	1.22
74	B-52PEN	* B-52PEN	233.471	0.59	1.81
75	MMIII	* TRID5	236.468	0.60	2.41
76	B1B	* ATB	252.016	0.64	3.04
77	B1B	* B1B	318.929	0.81	3.85
78	MMII	* MMII	360.030	0.91	4.76
79	MMIII		382.887	0.97	5.73
80	FB111PEN	* FB111PEN	473.731	1.20	6.93
81	B-52CMC	* B-52CMC	704.153	1.78	8.71
82	PKEEP	* TRID5	724.956	1.83	10.54

Parameters Of Y2 Sorted In Ascending Order
By Their Sum Of Squares

Number	Variable		Type I Sum of Squares	Percent Of SSR	Summed Percent Of SSR
83	POSC3	* POSC3	785.195	1.99	12.53
84	B1B		908.269	2.30	14.83
85	ATB	* ATB	1053.074	2.66	17.49
86	MICBM	* MICBM	1272.815	3.22	20.71
87	PKEEP		3272.555	8.28	28.99
88	TRID5	* TRID5	4475.646	11.32	40.31
89	ATB		7338.278	18.56	58.87
90	TRID5		16261.032	41.13	100.00

For Y2: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
1	88.60	84.63	3.97
2	61.21	63.13	-1.92
3	61.21	63.13	-1.92
4	88.69	84.64	4.05
5	88.60	88.83	-0.23
6	88.67	88.87	-0.20
7	88.60	87.98	0.62
8	88.60	86.57	2.03
9	61.21	59.71	1.50
10	61.21	59.72	1.49
11	88.65	86.60	2.05
12	88.60	87.95	0.65
13	42.99	45.69	-2.70
14	42.99	45.74	-2.75
15	48.64	51.99	-3.35
16	48.64	51.94	-3.30
17	89.41	89.20	0.21
18	89.43	89.19	0.24
19	89.43	89.21	0.22
20	89.53	89.23	0.30
21	77.89	78.95	-1.06
22	77.89	78.94	-1.05
23	69.77	70.02	-0.25
24	89.43	88.35	1.08
25	89.40	88.37	1.03
26	89.43	88.35	1.08
27	89.36	88.34	1.02
28	69.77	69.98	-0.21
29	69.77	70.00	-0.23
30	69.77	70.04	-0.27

For Y2: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
31	77.89	78.94	-1.05
32	77.89	78.94	-1.05
33	89.22	89.80	-0.58
34	89.22	89.82	-0.60
35	75.78	73.73	2.05
36	75.78	73.73	2.05
37	69.83	69.29	0.54
38	89.68	90.04	-0.36
39	88.03	90.07	-2.04
40	87.57	88.09	-0.52
41	87.57	88.11	-0.54
42	45.44	45.10	0.34
43	45.44	45.09	0.35
44	42.99	42.39	0.60
45	42.99	42.41	0.58
46	88.03	90.10	-2.07
47	89.74	90.07	-0.33
48	69.83	69.31	0.52
49	88.53	89.22	-0.69
50	88.53	89.21	-0.68
51	88.53	89.22	-0.69
52	88.53	89.22	-0.69
53	88.53	89.21	-0.68
54	88.53	89.21	-0.68
55	88.67	87.98	0.69
56	88.67	87.99	0.68
57	88.67	88.00	0.67
58	88.67	88.00	0.67
59	88.67	88.00	0.67
60	88.67	87.98	0.69
61	88.67	87.98	0.69
62	88.67	87.98	0.69
63	88.53	89.20	-0.67
64	88.53	89.20	-0.67
65	89.14	89.04	0.10
66	81.41	81.53	-0.12
67	81.42	81.51	-0.09
68	89.17	89.05	0.12
69	89.17	89.06	0.11
70	89.14	89.02	0.12
71	89.05	89.00	0.05
72	89.14	89.04	0.10
73	81.40	81.50	-0.10
74	81.42	81.50	-0.08
75	89.18	89.09	0.09
76	89.17	89.06	0.11
77	81.42	81.50	-0.08
78	81.40	81.48	-0.08

For Y2: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
79	81.41	81.54	-0.13
80	81.42	81.53	-0.11
81	89.46	89.95	-0.49
82	68.03	68.52	-0.49
83	67.97	68.50	-0.53
84	89.46	89.94	-0.48
85	58.59	58.07	0.52
86	58.56	58.05	0.51
87	58.56	58.03	0.53
88	89.41	89.92	-0.51
89	68.01	68.53	-0.52
90	68.01	68.51	-0.50
91	89.41	89.92	-0.51
92	58.54	58.05	0.49
93	37.13	36.64	0.49
94	37.13	36.64	0.49
95	37.13	36.63	0.50
96	37.13	36.63	0.50
97	89.01	88.86	0.15
98	81.29	81.35	-0.06
99	81.34	81.37	-0.03
100	89.06	88.89	0.17
101	61.62	59.69	1.93
102	61.61	59.68	1.93
103	53.62	54.88	-1.26
104	89.01	89.42	-0.41
105	81.24	81.89	-0.65
106	81.24	81.88	-0.64
107	89.01	89.42	-0.41
108	53.64	54.86	-1.22
109	45.87	47.30	-1.43
110	45.87	47.33	-1.46
111	53.85	52.15	1.70
112	53.85	52.15	1.70
113	89.58	90.06	-0.48
114	68.14	68.64	-0.50
115	68.14	68.65	-0.51
116	89.57	90.06	-0.49
117	89.58	90.06	-0.48
118	89.54	90.03	-0.49
119	85.66	85.16	0.50
120	85.66	85.15	0.51
121	64.21	63.73	0.48
122	64.24	63.74	0.50
123	85.66	85.15	0.51
124	85.68	85.18	0.50
125	64.24	63.77	0.47
126	64.24	63.74	0.50

For Y2: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
127	68.10	68.61	-0.51
128	68.14	68.65	-0.51
129	89.40	89.94	-0.54
130	89.40	89.94	-0.54
131	89.40	89.95	-0.55
132	89.40	89.97	-0.57
133	77.89	77.57	0.32
134	77.85	77.55	0.30
135	74.02	73.48	0.54
136	89.40	89.72	-0.32
137	89.40	89.71	-0.31
138	89.41	89.72	-0.31
139	89.43	89.75	-0.32
140	74.06	73.51	0.55
141	74.06	73.51	0.55
142	74.06	73.50	0.56
143	77.89	77.58	0.31
144	77.90	77.58	0.32
145	88.91	88.73	0.18
146	81.18	81.20	-0.02
147	81.14	81.18	-0.04
148	88.91	88.71	0.20
149	89.37	90.36	-0.99
150	89.40	90.37	-0.97
151	82.63	82.60	0.03
152	87.93	86.71	1.22
153	80.14	79.16	0.98
154	80.14	79.16	0.98
155	87.91	86.72	1.19
156	82.65	82.62	0.03
157	74.88	75.08	-0.20
158	74.88	75.07	-0.19
159	81.64	82.85	-1.21
160	81.64	82.84	-1.20
161	89.40	90.42	-1.02
162	76.13	77.04	-0.91
163	61.41	61.64	-0.23
164	90.62	90.90	-0.28
165	90.65	90.86	-0.21
166	89.35	90.42	-1.07
167	76.60	76.27	0.33
168	76.60	76.18	0.42
169	62.94	62.69	0.25
170	48.22	47.19	1.03
171	77.44	76.55	0.89
172	77.42	76.61	0.81
173	48.22	47.16	1.06
174	62.94	62.70	0.24

For Y2: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
175	76.16	76.96	-0.80
176	61.00	61.52	-0.52
177	89.31	90.01	-0.70
178	67.91	68.59	-0.68
179	67.91	68.60	-0.69
180	89.31	89.99	-0.68
181	89.77	90.01	-0.24
182	89.81	90.03	-0.22
183	87.32	86.65	0.67
184	89.35	89.11	0.24
185	67.91	67.68	0.23
186	67.91	67.68	0.23
187	89.31	89.08	0.23
188	87.32	86.63	0.69
189	65.90	65.22	0.68
190	65.90	65.21	0.69
191	68.37	68.60	-0.23
192	68.37	68.61	-0.24
193	88.76	88.76	0.00

Regression Output for Dependent Variable: Y3

Model Sum of Squares (SSR)	Error Sum of Squares (SSE)	Total Sum of Squares (SSTO)	R-Square
1243.54	172.89	1416.43	0.8779

Parameters Of Y3 Sorted In Ascending Order By Their Sum Of Squares

Number	Variable	Type I Sum of Squares	Percent Of SSR	Summed Percent Of SSR
1	POSTRIC4 * FB111PEN	0.003	0.00	0.00
2	MICBM * POSTRIC4	0.004	0.00	0.00
3	MMII * PKEEP	0.006	0.00	0.00
4	B-52CMC * FB111PEN	0.010	0.00	0.00
5	MMII * FB111PEN	0.025	0.00	0.00
6	MMIII * MMIII	0.028	0.00	0.01
7	MICBM * TRID5	0.029	0.00	0.01
8	B-52CMC * B-52CMC	0.044	0.00	0.01
9	MMII * POSC3	0.046	0.00	0.02
10	MMII * B-52CMC	0.052	0.00	0.02
11	MMIII * POSC3	0.065	0.01	0.02
12	PKEEP * B1B	0.066	0.01	0.03
13	POSC3 * TRID5	0.122	0.01	0.04
14	MMIII * B-52PEN	0.123	0.01	0.05
15	POSC3 * ATB	0.128	0.01	0.06
16	POSTRIC4 * TRID5	0.146	0.01	0.07
17	MMIII * FB111PEN	0.164	0.01	0.09
18	MMII * B1B	0.205	0.02	0.10
19	MMII * B-52PEN	0.263	0.02	0.12
20	FB111PEN	0.312	0.03	0.15
21	MICBM * B-52CMC	0.314	0.03	0.17
22	TRID5 * B-52PEN	0.328	0.03	0.20
23	MMIII * ATB	0.336	0.03	0.23
24	B1B * B1B	0.342	0.03	0.25
25	B-52PEN * B1B	0.345	0.03	0.28
26	B-52PEN * ATB	0.360	0.03	0.31
27	PKEEP * MICBM	0.416	0.03	0.34
28	PKEEP * FB111PEN	0.473	0.04	0.38
29	PKEEP * POSC3	0.574	0.05	0.43
30	MMII * TRID5	0.575	0.05	0.47
31	POSTRIC4 * B1B	0.605	0.05	0.52
32	POSTRIC4 * B-52CMC	0.605	0.05	0.57
33	FB111PEN * ATB	0.620	0.05	0.62
34	MMII * POSTRIC4	0.664	0.05	0.68
35	MMII * MICBM	0.664	0.05	0.73
36	TRID5 * B1B	0.710	0.06	0.79

Parameters Of Y3 Sorted In Ascending Order
By Their Sum Of Squares

Number	Variable	Type I Sum of Squares	Percent Of SSR	Summed Percent Of SSR
37	POSC3 * FB111PEN	0.718	0.06	0.84
38	TRID5 * FB111PEN	0.770	0.06	0.91
39	MICBM * POSC3	0.815	0.07	0.97
40	POSC3 * B-52PEN	0.842	0.07	1.04
41	POSTRIC4 * ATB	0.894	0.07	1.11
42	POSC3 * B1B	0.925	0.07	1.18
43	MICBM	0.958	0.08	1.26
44	MMII * MMIII	0.980	0.08	1.34
45	B-52PEN * FB111PEN	1.097	0.09	1.43
46	B-52CMC	1.216	0.10	1.53
47	MICBM * B-52PEN	1.216	0.10	1.62
48	POSTRIC4 * B-52PEN	1.254	0.10	1.73
49	MICBM * FB111PEN	1.415	0.11	1.84
50	MMIII * POSTRIC4	1.452	0.12	1.96
51	MMII * ATB	1.563	0.13	2.08
52	B-52PEN * B-52CMC	1.690	0.14	2.22
53	MMIII * MICBM	1.703	0.14	2.35
54	FB111PEN * B1B	2.250	0.18	2.54
55	TRID5 * ATB	2.379	0.19	2.73
56	MICBM * B1B	2.496	0.20	2.93
57	MMII	2.592	0.21	3.14
58	PKEEP * ATB	3.106	0.25	3.39
59	B-52PEN * B-52PEN	3.341	0.27	3.65
60	POSC3 * POSTRIC4	3.450	0.28	3.93
61	FB111PEN * FB111PEN	4.960	0.40	4.33
62	MMII * MMII	5.118	0.41	4.74
63	B1B * ATB	5.534	0.45	5.19
64	POSC3	5.814	0.47	5.65
65	POSC3 * B-52CMC	5.917	0.48	6.13
66	PKEEP * PKEEP	6.599	0.53	6.66
67	MICBM * ATB	7.480	0.60	7.26
68	PKEEP * B-52PEN	7.535	0.61	7.87
69	MMIII * B1B	8.556	0.69	8.56
70	PKEEP * POSTRIC4	8.940	0.72	9.28
71	MMIII * TRID5	9.030	0.73	10.00
72	ATB * ATB	11.209	0.90	10.90
73	ATB	11.972	0.96	11.87
74	B-52PEN	13.377	1.08	12.94
75	PKEEP * B-52CMC	13.794	1.11	14.05
76	MMIII * B-52CMC	15.602	1.25	15.31
77	POSTRIC4 * POSTRIC4	16.403	1.32	16.62
78	B-52CMC * ATB	17.935	1.44	18.07
79	PKEEP * TRID5	20.070	1.61	19.68
80	MICBM * MICBM	30.349	2.44	22.12
81	MMIII * PKEEP	33.178	2.67	24.79
82	POSC3 * POSC3	33.447	2.69	27.48

Parameters Of Y3 Sorted In Ascending Order
By Their Sum Of Squares

Number	Variable		Type I Sum of Squares	Percent Of SSR	Summed Percent Of SSR
83	MMIII		34.810	2.80	30.28
84	POSTRIC4		37.119	2.98	33.26
85	B-52CMC	* B1B	41.441	3.33	36.60
86	B1B		50.321	4.05	40.64
87	TRID5	* B-52CMC	51.051	4.11	44.75
88	TRID5	* TRID5	95.946	7.72	52.46
89	PKBEP		130.074	10.46	62.92
90	TRID5		461.068	37.08	100.00

For Y3: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
1	81.69	82.87	-1.18
2	76.16	75.91	0.25
3	74.65	74.90	-0.25
4	81.88	82.21	-0.33
5	80.28	79.86	0.42
6	80.24	80.26	-0.02
7	79.13	78.99	0.14
8	82.33	82.59	-0.26
9	75.54	76.16	-0.62
10	75.50	75.37	0.13
11	82.33	82.14	0.19
12	78.75	78.80	-0.05
13	76.01	75.03	0.98
14	76.01	75.57	0.44
15	76.01	76.31	-0.30
16	76.01	75.55	0.46
17	85.71	84.20	1.51
18	81.43	81.23	0.20
19	78.52	78.17	0.35
20	80.19	80.02	0.17
21	74.77	74.41	0.36
22	75.92	75.60	0.32
23	75.74	76.28	-0.54
24	78.77	79.12	-0.35
25	76.43	76.40	0.03
26	74.41	74.54	-0.13
27	76.03	76.14	-0.11
28	75.70	76.29	-0.59
29	76.32	77.44	-1.12
30	76.32	76.30	0.02

For Y3: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
31	75.30	75.38	-0.08
32	75.26	75.31	-0.05
33	83.41	86.22	-2.81
34	83.29	81.89	1.40
35	78.96	77.94	1.02
36	73.84	75.13	-1.29
37	73.84	75.40	-1.56
38	86.04	86.67	-0.63
39	79.56	79.94	-0.38
40	80.73	80.14	0.59
41	78.11	78.44	-0.33
42	78.96	78.97	-0.01
43	73.84	73.53	0.31
44	73.65	73.15	0.50
45	78.96	78.04	0.92
46	78.33	77.68	0.65
47	82.10	81.78	0.32
48	78.96	77.66	1.30
49	77.85	78.06	-0.21
50	75.20	76.55	-1.35
51	75.68	75.15	0.53
52	74.57	75.61	-1.04
53	74.53	74.59	-0.06
54	74.35	76.12	-1.77
55	75.13	75.36	-0.23
56	75.18	76.39	-1.21
57	76.31	75.72	0.59
58	76.89	75.42	1.47
59	75.75	75.04	0.71
60	75.75	74.93	0.82
61	76.89	76.16	0.73
62	76.27	75.54	0.73
63	75.06	75.46	-0.40
64	75.66	74.98	0.68
65	81.05	80.90	0.15
66	73.74	75.18	-1.44
67	78.23	77.07	1.16
68	76.73	76.35	0.38
69	76.73	75.76	0.97
70	81.92	79.53	2.39
71	73.61	76.44	-2.83
72	80.17	79.67	0.50
73	73.47	74.63	-1.16
74	78.23	78.95	-0.72
75	76.73	77.55	-0.82
76	76.73	75.10	1.63
77	76.94	77.28	-0.34
78	73.47	72.18	1.29

For Y3: Residuals Associated With The Design Points

Design Point	ABM's Response	Postulated Model's Response	Residual
79	73.74	74.59	-0.85
80	76.94	77.25	-0.31
81	81.88	83.37	-1.49
82	84.88	82.55	2.33
83	84.65	82.39	2.26
84	81.11	82.43	-1.32
85	77.88	78.08	-0.20
86	77.87	78.14	-0.27
87	74.94	76.51	-1.57
88	81.66	81.35	0.31
89	81.01	81.20	-0.19
90	80.57	80.98	-0.41
91	81.33	80.35	0.98
92	74.44	76.38	-1.94
93	76.58	75.48	1.10
94	75.44	74.81	0.63
95	75.49	75.78	-0.29
96	76.58	76.50	0.08
97	82.63	83.73	-1.10
98	83.20	81.47	1.73
99	80.02	80.88	-0.86
100	82.62	82.56	0.06
101	74.98	76.21	-1.23
102	78.60	77.96	0.64
103	78.60	77.80	0.80
104	82.15	83.03	-0.88
105	81.40	80.32	1.08
106	79.66	80.24	-0.58
107	82.75	82.37	0.38
108	74.98	76.56	-1.58
109	76.32	75.27	1.05
110	76.01	75.93	0.08
111	76.01	76.54	-0.53
112	76.32	75.38	0.94
113	83.05	82.17	0.88
114	79.79	78.59	1.20
115	76.62	78.36	-1.74
116	77.01	77.70	-0.69
117	76.90	77.01	-0.11
118	79.87	80.18	-0.31
119	78.81	76.56	2.25
120	79.41	78.30	1.11
121	76.51	75.31	1.20
122	78.83	79.03	-0.20
123	77.29	77.79	-0.50
124	75.06	77.35	-2.29
125	78.81	79.19	-0.38
126	73.77	74.17	-0.40

For Y3: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
127	78.79	77.21	1.58
128	76.66	78.27	-1.61
129	80.82	79.56	1.26
130	79.60	79.79	-0.19
131	78.14	78.92	-0.78
132	78.10	78.59	-0.49
133	78.27	77.35	0.92
134	75.87	75.70	0.17
135	75.15	75.53	-0.38
136	79.80	79.31	0.49
137	79.84	79.39	0.45
138	76.74	78.28	-1.54
139	78.17	78.11	0.06
140	78.30	76.95	1.35
141	76.55	76.44	0.11
142	74.03	74.92	-0.89
143	74.03	75.25	-1.22
144	77.66	77.00	0.66
145	79.05	79.97	-0.92
146	78.36	76.77	1.59
147	77.24	77.17	0.07
148	78.99	78.86	0.13
149	79.37	79.18	0.19
150	79.22	79.44	-0.22
151	77.14	77.36	-0.22
152	77.15	76.58	0.57
153	75.75	76.31	-0.56
154	76.89	76.30	0.59
155	75.25	75.07	0.18
156	75.25	76.69	-1.44
157	76.89	76.34	0.55
158	75.75	75.51	0.24
159	74.34	74.67	-0.33
160	75.48	75.90	-0.42
161	81.59	82.28	-0.69
162	79.78	79.54	0.24
163	78.52	78.47	0.05
164	79.72	78.86	0.86
165	78.85	78.40	0.45
166	81.62	81.14	0.48
167	78.53	77.65	0.88
168	78.96	78.04	0.92
169	76.05	77.06	-1.01
170	76.32	76.25	0.07
171	74.05	74.88	-0.83
172	74.00	75.17	-1.17
173	76.32	76.19	0.13
174	76.01	76.31	-0.30

For Y3: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
175	79.79	78.03	1.76
176	75.80	77.65	-1.85
177	81.10	79.76	1.34
178	76.81	76.57	0.24
179	74.55	74.94	-0.39
180	78.30	77.46	0.84
181	75.62	75.47	0.15
182	77.99	77.71	0.28
183	77.70	76.68	1.02
184	78.35	77.92	0.43
185	75.38	75.97	-0.59
186	75.33	75.16	0.17
187	77.50	76.44	1.06
188	74.39	75.26	-0.87
189	76.01	76.72	-0.71
190	76.01	77.47	-1.46
191	75.74	77.25	-1.51
192	75.70	75.69	0.01
193	75.51	75.51	0.00

Regression Output for Dependent Variable: Y4

Model Sum of Squares (SSR)	Error Sum of Squares (SSE)	Total Sum of Squares (SSTO)	R-Square
31638.00	2058.35	33696.35	0.9389

Parameters Of Y4 Sorted In Ascending Order By Their Sum Of Squares

Number	Variable		Type I Sum of Squares	Percent Of SSR	Summed Percent Of SSR
1	MMII	* B-52CMC	0.000	0.00	0.00
2	MMII	* FB111PEN	0.000	0.00	0.00
3	MMII	* B1B	0.004	0.00	0.00
4	MICBM	* B-52CMC	0.098	0.00	0.00
5	PKEEP	* MICBM	0.117	0.00	0.00
6	MMII	* B-52PEN	0.194	0.00	0.00
7	MICBM	* B1B	0.483	0.00	0.00
8	POSTRIC4	* FB111PEN	0.753	0.00	0.01
9	MMII	* PKEEP	1.118	0.00	0.01
10	MMII	* TRID5	1.288	0.00	0.01
11	MMII	* MICBM	2.095	0.01	0.02
12	B-52CMC	* FB111PEN	2.379	0.01	0.03
13	B1B	* ATB	2.890	0.01	0.04
14	MMIII	* FB111PEN	3.349	0.01	0.05
15	POSTRIC4	* POSTRIC4	3.582	0.01	0.06
16	MMII	* POSC3	3.822	0.01	0.07
17	MICBM	* TRID5	4.633	0.01	0.08
18	B-52PEN	* FB111PEN	4.917	0.02	0.10
19	FB111PEN	* ATB	7.715	0.02	0.12
20	MICBM	* FB111PEN	8.171	0.03	0.15
21	MMII	* POSTRIC4	8.512	0.03	0.18
22	PKEEP	* FB111PEN	10.224	0.03	0.21
23	TRID5	* FB111PEN	13.123	0.04	0.25
24	MMIII	* TRID5	13.579	0.04	0.29
25	MMII	* MMIII	14.251	0.05	0.34
26	MICBM	* POSTRIC4	14.383	0.05	0.38
27	MMIII	* MICBM	15.288	0.05	0.43
28	MMIII	* PKEEP	15.701	0.05	0.48
29	MMII	* ATB	19.250	0.06	0.54
30	MICBM	* B-52PEN	19.780	0.06	0.61
31	MMIII	* POSC3	19.803	0.06	0.67
32	POSC3	* FB111PEN	21.045	0.07	0.74
33	FB111PEN	* B1B	22.326	0.07	0.81
34	MMIII	* POSTRIC4	24.626	0.08	0.88
35	PKEEP	* PKEEP	25.120	0.08	0.96
36	POSC3	* ATB	30.195	0.10	1.06

Parameters Of Y4 Sorted In Ascending Order
By Their Sum Of Squares

Number	Variable		Type I Sum of Squares	Percent Of SSR	Summed Percent Of SSR
37	B-52PEN	* B1B	30.305	0.10	1.15
38	MMIII	* B-52PEN	33.252	0.11	1.26
39	MICBM	* ATB	42.023	0.13	1.39
40	MMIII	* MMIII	44.002	0.14	1.53
41	B-52PEN	* ATB	46.547	0.15	1.68
42	PKEEP	* B1B	46.717	0.15	1.83
43	B-52PEN	* B-52CMC	47.231	0.15	1.98
44	MICBM	* POSC3	47.990	0.15	2.13
45	PKEEP	* TRID5	50.304	0.16	2.29
46	POSTRIC4	* B1B	54.575	0.17	2.46
47	POSC3	* POSTRIC4	54.871	0.17	2.63
48	POSTRIC4	* B-52PEN	55.094	0.17	2.81
49	MMII		59.367	0.19	2.99
50	POSC3	* B1B	61.466	0.19	3.19
51	PKEEP	* B-52PEN	72.293	0.23	3.42
52	POSTRIC4	* B-52CMC	79.255	0.25	3.67
53	MMIII	* ATB	81.857	0.26	3.93
54	TRID5	* B-52PEN	81.993	0.26	4.18
55	POSTRIC4	* TRID5	85.701	0.27	4.46
56	B-52PEN	* B-52PEN	99.317	0.31	4.77
57	POSC3	* B-52PEN	107.381	0.34	5.11
58	MMIII	* B-52CMC	112.307	0.35	5.46
59	TRID5	* ATB	113.796	0.36	5.82
60	PKEEP	* ATB	116.532	0.37	6.19
61	PKEEP	* POSC3	128.142	0.41	6.60
62	PKEEP	* POSTRIC4	128.312	0.41	7.00
63	B1B	* B1B	130.637	0.41	7.42
64	MICBM	* MICBM	131.454	0.42	7.83
65	POSC3	* B-52CMC	131.733	0.42	8.25
66	FB111PEN	* FB111PEN	134.232	0.42	8.67
67	MMII	* MMII	152.067	0.48	9.15
68	B-52CMC	* ATB	182.048	0.58	9.73
69	TRID5	* B1B	195.720	0.62	10.35
70	POSC3	* TRID5	201.498	0.64	10.98
71	FB111PEN		208.117	0.66	11.64
72	POSTRIC4	* ATB	219.661	0.69	12.34
73	B-52CMC	* B1B	224.925	0.71	13.05
74	MMIII	* B1B	226.051	0.71	13.76
75	ATB	* ATB	254.790	0.81	14.57
76	PKEEP	* B-52CMC	278.362	0.88	15.45
77	POSC3	* POSC3	279.640	0.88	16.33
78	MICBM		327.565	1.04	17.37
79	B-52CMC	* B-52CMC	499.011	1.58	18.94
80	TRID5	* TRID5	527.313	1.67	20.61
81	TRID5	* B-52CMC	725.090	2.29	22.90
82	MMIII		1235.347	3.90	26.81

Parameters Of Y4 Sorted In Ascending Order
By Their Sum Of Squares

Number	Variable	Type I Sum of Squares	Percent Of SSR	Summed Percent Of SSR
83	B-52PEN	1362.440	4.31	31.11
84	B1B	1792.358	5.67	36.78
85	ATB	1959.722	6.19	42.97
86	PKEEP	2199.727	6.95	49.92
87	POSTRIC4	2581.148	8.16	58.08
88	B-52CMC	3720.085	11.76	69.84
89	POSC3	3830.991	12.11	81.95
90	TRID5	5710.825	18.05	100.00

For Y4: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
1	86.93	88.47	-1.54
2	75.93	78.92	-2.99
3	63.67	59.55	4.12
4	79.82	83.30	-3.48
5	74.79	72.24	2.55
6	82.21	81.86	0.35
7	80.78	83.20	-2.42
8	85.08	86.04	-0.96
9	73.55	75.68	-2.13
10	52.32	54.36	-2.04
11	78.28	78.91	-0.63
12	74.20	71.63	2.57
13	45.35	43.39	1.96
14	70.95	69.17	1.78
15	70.95	68.63	2.32
16	45.35	44.81	0.54
17	80.67	79.30	1.37
18	79.06	80.07	-1.01
19	73.72	71.81	1.91
20	76.14	78.46	-2.32
21	67.77	67.30	0.47
22	76.82	79.47	-2.65
23	73.78	73.22	0.56
24	77.49	77.01	0.48
25	74.97	73.71	1.26
26	60.66	60.48	0.18
27	72.44	71.21	1.23
28	55.37	56.09	-0.72
29	37.22	36.86	0.36
30	61.26	61.42	-0.16

For Y4: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
31	72.98	71.74	1.24
32	49.94	52.15	-2.21
33	86.59	86.51	0.08
34	82.31	90.47	-8.16
35	42.13	53.49	-11.36
36	72.53	76.46	-3.93
37	72.53	72.68	-0.15
38	83.00	80.58	2.42
39	77.43	71.38	6.05
40	78.77	76.97	1.80
41	75.37	69.13	6.24
42	42.13	39.24	2.89
43	72.53	74.01	-1.48
44	72.64	70.57	2.07
45	42.13	36.12	6.01
46	73.09	63.85	9.24
47	81.02	84.85	-3.83
48	42.13	50.03	-7.90
49	76.88	70.78	6.10
50	76.89	71.58	5.31
51	72.27	68.65	3.62
52	74.71	70.06	4.65
53	54.68	55.17	-0.49
54	71.84	66.24	5.60
55	58.45	61.49	-3.04
56	75.00	72.95	2.05
57	72.64	71.74	0.90
58	58.18	64.36	-6.18
59	66.02	67.79	-1.77
60	40.24	45.96	-5.72
61	32.26	37.94	-5.68
62	50.46	55.69	-5.23
63	64.81	62.46	2.35
64	46.70	49.17	-2.47
65	80.87	75.18	5.69
66	79.57	78.56	1.01
67	67.73	66.00	1.73
68	77.71	77.62	0.09
69	67.69	67.86	-0.17
70	72.91	74.32	-1.41
71	67.03	63.66	3.37
72	73.93	71.92	2.01
73	72.64	69.75	2.89
74	42.03	45.72	-3.69
75	65.92	62.88	3.04
76	41.99	45.72	-3.73
77	18.19	21.17	-2.98
78	48.91	54.11	-5.20

For Y4: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
79	72.38	70.32	2.06
80	44.13	48.86	-4.73
81	88.04	87.76	0.28
82	81.88	88.65	-6.77
83	80.92	85.90	-4.98
84	87.55	87.78	-0.23
85	76.32	77.04	-0.72
86	77.77	80.64	-2.87
87	72.64	68.98	3.66
88	82.55	85.36	-2.81
89	77.59	75.77	1.82
90	76.35	72.16	4.19
91	81.27	84.52	-3.25
92	70.20	64.52	5.68
93	42.30	41.49	0.81
94	50.29	48.73	1.56
95	72.79	70.87	1.92
96	66.21	64.49	1.72
97	84.82	83.85	0.97
98	80.70	83.05	-2.35
99	78.45	75.81	2.64
100	81.38	82.12	-0.74
101	67.60	66.10	1.50
102	73.15	76.88	-3.73
103	72.95	74.81	-1.86
104	83.36	82.58	0.78
105	79.71	81.71	-2.00
106	76.99	74.04	2.95
107	79.63	80.41	-0.78
108	66.43	63.58	2.85
109	43.03	43.22	-0.19
110	61.19	59.95	1.24
111	62.37	62.09	0.28
112	44.24	45.80	-1.56
113	80.07	78.25	1.82
114	75.88	81.86	-5.98
115	72.94	68.60	4.34
116	78.28	78.48	-0.20
117	74.46	71.27	3.19
118	77.73	77.91	-0.18
119	73.56	76.90	-3.34
120	77.05	81.32	-4.27
121	72.95	75.89	-2.94
122	51.21	52.03	-0.82
123	73.20	70.96	2.24
124	65.39	59.67	5.72
125	33.09	33.92	-0.83
126	64.59	64.65	-0.06

For Y4: Residuals Associated With The Design Points

Design Point	ABM's Response	Postulated Model's Response	Residual
127	72.71	74.70	-1.99
128	57.84	54.57	3.27
129	78.28	79.06	-0.78
130	77.75	77.81	-0.06
131	73.83	67.69	6.14
132	75.02	70.48	4.54
133	48.09	53.92	-5.83
134	72.56	74.30	-1.74
135	72.55	72.89	-0.34
136	77.02	76.59	0.43
137	75.59	75.36	0.23
138	72.59	65.24	7.35
139	72.97	68.02	4.95
140	46.79	52.52	-5.73
141	41.72	46.55	-4.83
142	69.53	68.47	1.06
143	70.70	69.86	0.84
144	41.72	47.94	-6.22
145	77.68	76.16	1.52
146	73.08	75.80	-2.72
147	72.59	71.76	0.83
148	76.30	76.84	-0.54
149	73.04	69.70	3.34
150	74.51	71.04	3.47
151	72.63	72.64	-0.01
152	72.55	73.85	-1.30
153	56.10	58.46	-2.36
154	48.12	52.58	-4.46
155	71.36	72.70	-1.34
156	71.36	69.47	1.89
157	48.12	48.65	-0.53
158	56.10	56.55	-0.45
159	72.15	69.98	2.17
160	64.39	63.92	0.47
161	86.16	86.19	-0.03
162	80.30	82.41	-2.11
163	77.87	78.87	-1.00
164	80.86	80.94	-0.08
165	73.93	71.10	2.83
166	79.18	81.89	-2.71
167	73.71	73.32	0.39
168	80.36	88.94	-8.58
169	73.32	74.37	-1.05
170	64.92	63.99	0.93
171	75.40	76.86	-1.46
172	58.42	55.70	2.72
173	39.14	37.34	1.80
174	51.56	53.26	-1.70

For Y4: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
175	73.47	72.62	0.85
176	72.72	63.54	9.18
177	79.48	78.21	1.27
178	75.90	77.82	-1.92
179	66.17	63.23	2.94
180	73.73	74.10	-0.37
181	72.55	70.20	2.35
182	78.06	78.10	-0.04
183	77.18	80.55	-3.37
184	78.92	79.21	-0.29
185	74.07	74.43	-0.36
186	54.98	56.93	-1.95
187	72.47	72.19	0.28
188	72.08	69.73	2.35
189	46.29	47.99	-1.70
190	70.09	69.29	0.80
191	72.93	71.22	1.71
192	51.15	52.85	-1.70
193	64.99	64.99	0.00

Regression Output for Dependent Variable: Y5

Model Sum of Squares (SSR)	Error Sum of Squares (SSE)	Total Sum of Squares (SSTO)	R-Square
10786.68	267.84	11054.52	0.9758

Parameters Of Y5 Sorted In Ascending Order By Their Sum Of Squares

Number	Variable			Type I Sum of Squares	Percent Of SSR	Summed Percent Of SSR
1	MICBM	*	B-52CMC	0.000	0.00	0.00
2	MMII	*	FB111PEN	0.001	0.00	0.00
3	MMII	*	B-52CMC	0.004	0.00	0.00
4	PKEEP	*	MICBM	0.008	0.00	0.00
5	MMII	*	B-52PEN	0.012	0.00	0.00
6	MMII	*	B1B	0.014	0.00	0.00
7	MMII	*	MICBM	0.022	0.00	0.00
8	POSTRIC4	*	FB111PEN	0.076	0.00	0.00
9	MMII	*	MMIII	0.160	0.00	0.00
10	MMII	*	PKEEP	0.207	0.00	0.00
11	ATB	*	ATB	0.266	0.00	0.01
12	MMIII	*	MICBM	0.278	0.00	0.01
13	MICBM	*	TRID5	0.302	0.00	0.01
14	POSTRIC4	*	POSTRIC4	0.332	0.00	0.02
15	B-52CMC	*	FB111PEN	0.456	0.00	0.02
16	FB111PEN	*	FB111PEN	0.461	0.00	0.02
17	B-52PEN	*	FB111PEN	0.494	0.00	0.03
18	MICBM	*	B1B	0.515	0.00	0.03
19	MMII	*	POSC3	0.697	0.01	0.04
20	MMIII	*	FB111PEN	0.710	0.01	0.05
21	B1B	*	B1B	0.770	0.01	0.05
22	MICBM	*	FB111PEN	0.852	0.01	0.06
23	FB111PEN	*	ATB	0.980	0.01	0.07
24	MMII	*	POSTRIC4	1.108	0.01	0.08
25	B-52PEN	*	B-52PEN	1.185	0.01	0.09
26	TRID5	*	FB111PEN	1.850	0.02	0.11
27	PKEEP	*	FB111PEN	2.103	0.02	0.13
28	MMII	*	ATB	2.110	0.02	0.15
29	MICBM	*	MICBM	2.410	0.02	0.17
30	MICBM	*	B-52PEN	2.681	0.02	0.20
31	MICBM	*	POSTRIC4	2.698	0.03	0.22
32	FB111PEN	*	B1B	2.781	0.03	0.25
33	MMII	*	MMII	2.841	0.03	0.27
34	POSC3	*	FB111PEN	2.848	0.03	0.30
35	MMIII	*	POSC3	3.133	0.03	0.33
36	MMIII	*	POSTRIC4	3.930	0.04	0.36

Parameters Of Y5 Sorted In Ascending Order
By Their Sum Of Squares

Number	Variable	Type I Sum of Squares	Percent Of SSR	Summed Percent Of SSR
37	B-52PEN * B1B	3.990	0.04	0.40
38	MMII * TRID5	4.083	0.04	0.44
39	MICBM * ATB	4.337	0.04	0.48
40	B1B * ATB	4.484	0.04	0.52
41	PKEEP * TRID5	4.709	0.04	0.56
42	MMIII * B-52PEN	5.437	0.05	0.61
43	MMIII * PKEEP	5.605	0.05	0.67
44	POSC3 * POSC3	5.656	0.05	0.72
45	B-52PEN * B-52CMC	6.350	0.06	0.78
46	PKEEP * PKEEP	6.487	0.06	0.84
47	POSC3 * ATB	6.490	0.06	0.90
48	B-52PEN * ATB	7.023	0.07	0.96
49	MICBM * POSC3	7.089	0.07	1.03
50	TRID5 * TRID5	7.152	0.07	1.10
51	POSTRIC4 * B-52PEN	7.604	0.07	1.17
52	POSTRIC4 * B1B	7.756	0.07	1.24
53	PKEEP * B-52PEN	8.080	0.07	1.31
54	PKEEP * B1B	8.165	0.08	1.39
55	POSC3 * B1B	9.429	0.09	1.48
56	POSTRIC4 * B-52CMC	12.076	0.11	1.59
57	MMIII * B-52CMC	13.104	0.12	1.71
58	B-52CMC * B-52CMC	13.127	0.12	1.83
59	POSC3 * POSTRIC4	13.213	0.12	1.95
60	MMIII * ATB	14.592	0.14	2.09
61	TRID5 * B-52PEN	14.726	0.14	2.23
62	POSC3 * B-52PEN	14.803	0.14	2.36
63	POSTRIC4 * TRID5	15.366	0.14	2.51
64	PKEEP * ATB	16.261	0.15	2.66
65	MMIII * MMIII	16.453	0.15	2.81
66	POSC3 * B-52CMC	17.223	0.16	2.97
67	PKEEP * POSTRIC4	17.577	0.16	3.13
68	MMIII * TRID5	19.492	0.18	3.31
69	PKEEP * POSC3	21.229	0.20	3.51
70	B-52CMC * B1B	23.136	0.21	3.72
71	TRID5 * ATB	23.571	0.22	3.94
72	B-52CMC * ATB	23.668	0.22	4.16
73	MMIII * B1B	30.609	0.28	4.44
74	TRID5 * B1B	32.919	0.31	4.75
75	MMII	33.698	0.31	5.06
76	POSTRIC4 * ATB	34.466	0.32	5.38
77	PKEEP * B-52CMC	37.628	0.35	5.73
78	POSC3 * TRID5	38.192	0.35	6.08
79	FB111PEN	39.125	0.36	6.45
80	MICBM	74.025	0.69	7.13
81	TRID5 * B-52CMC	94.090	0.87	8.01
82	B-52PEN	253.805	2.35	10.36

Parameters Of Y5 Sorted In Ascending Order
By Their Sum Of Squares

Number	Variable	Type I Sum of Squares	Percent Of SSR	Summed Percent Of SSR
83	MMIII	446.900	4.14	14.50
84	POSTRIC4	541.260	5.02	19.52
85	B-52CMC	656.000	6.08	25.60
86	B1B	701.521	6.50	32.11
87	POSC3	733.597	6.80	38.91
88	PKEEP	1150.397	10.66	49.57
89	ATB	1448.373	13.43	63.00
90	TRID5	3991.239	37.00	100.00

For Y5: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
1	85.75	85.73	0.02
2	74.36	75.95	-1.59
3	68.57	67.39	1.18
4	82.90	83.36	-0.46
5	80.32	79.19	1.13
6	83.35	83.34	0.01
7	82.53	83.22	-0.69
8	85.01	85.22	-0.21
9	73.23	74.00	-0.77
10	64.12	64.61	-0.49
11	82.16	82.00	0.16
12	79.48	78.24	1.24
13	56.77	56.43	0.34
14	67.57	67.60	-0.03
15	69.47	69.15	0.32
16	58.67	58.82	-0.15
17	83.88	83.16	0.72
18	82.44	82.80	-0.36
19	79.43	78.70	0.73
20	81.03	81.81	-0.78
21	73.86	73.84	0.02
22	78.27	79.38	-1.11
23	74.90	74.73	0.17
24	81.25	80.88	0.37
25	79.58	78.87	0.71
26	72.99	72.78	0.21
27	78.11	77.54	0.57
28	66.84	67.21	-0.37
29	59.61	59.60	0.01
30	69.76	69.88	-0.12

For Y5: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
31	76.52	76.18	0.34
32	66.78	67.88	-1.10
33	86.08	86.33	-0.25
34	84.25	86.95	-2.70
35	63.71	67.67	-3.96
36	75.38	76.75	-1.37
37	74.12	74.37	-0.25
38	85.41	84.51	0.90
39	81.10	79.24	1.86
40	81.88	81.15	0.73
41	79.79	77.43	2.36
42	57.17	55.98	1.19
43	68.83	69.40	-0.57
44	67.53	66.93	0.60
45	55.86	53.50	2.36
46	78.97	75.51	3.46
47	83.59	85.12	-1.53
48	62.45	65.28	-2.83
49	80.80	78.48	2.32
50	80.16	78.43	1.73
51	78.33	76.84	1.49
52	79.09	77.59	1.50
53	70.63	71.07	-0.44
54	77.41	75.80	1.61
55	72.33	73.47	-1.14
56	79.60	78.80	0.80
57	78.84	78.11	0.73
58	72.92	74.88	-1.96
59	75.95	76.28	-0.33
60	65.07	67.09	-2.02
61	61.98	64.01	-2.03
62	69.23	71.08	-1.85
63	74.75	74.07	0.68
64	67.54	68.63	-1.09
65	83.37	81.13	2.24
66	79.47	79.39	0.08
67	75.37	74.40	0.97
68	81.13	80.94	0.19
69	76.76	76.60	0.16
70	80.34	80.26	0.08
71	75.53	74.83	0.70
72	80.34	79.34	1.00
73	76.29	75.43	0.86
74	64.53	66.28	-1.75
75	76.02	75.00	1.02
76	65.92	67.02	-1.10
77	54.30	55.52	-1.22
78	66.28	68.14	-1.86

For Y5: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
79	76.39	75.74	0.65
80	65.24	67.27	-2.03
81	86.48	86.84	-0.36
82	79.88	82.33	-2.45
83	79.38	81.09	-1.71
84	85.98	86.59	-0.61
85	74.40	74.50	-0.10
86	75.00	76.12	-1.12
87	71.84	70.55	1.29
88	83.84	85.20	-1.36
89	77.09	76.53	0.56
90	76.55	75.02	1.53
91	83.35	84.67	-1.32
92	70.66	68.67	1.99
93	54.97	54.15	0.82
94	58.06	57.03	1.03
95	67.65	66.74	0.91
96	65.05	64.14	0.91
97	85.18	85.09	0.09
98	81.68	82.39	-0.71
99	80.20	79.38	0.82
100	83.76	84.07	-0.31
101	70.65	69.95	0.70
102	73.83	74.79	-0.96
103	72.04	72.74	-0.70
104	84.45	84.46	-0.01
105	80.93	81.65	-0.72
106	79.38	78.53	0.85
107	83.03	83.34	-0.31
108	68.51	67.78	0.73
109	57.26	57.24	0.02
110	64.52	64.19	0.33
111	66.66	66.36	0.30
112	59.41	59.51	-0.10
113	83.45	82.68	0.77
114	76.56	78.84	-2.28
115	74.34	73.07	1.27
116	81.50	81.78	-0.28
117	79.87	78.69	1.18
118	81.96	82.11	-0.15
119	78.96	79.72	-0.76
120	80.57	81.94	-1.37
121	73.43	74.28	-0.85
122	64.53	64.89	-0.36
123	78.45	77.42	1.03
124	74.62	72.68	1.94
125	57.24	57.50	-0.26
126	69.35	69.40	-0.05

For Y5: Residuals Associated With The Design Points

Design Point	ABM's Response	Postulated Model's Response	Residual
127	74.92	75.61	-0.69
128	68.19	67.33	0.86
129	82.11	82.23	-0.12
130	81.54	81.74	-0.20
131	79.43	77.20	2.23
132	80.01	78.36	1.65
133	66.49	68.67	-2.18
134	76.30	76.87	-0.57
135	75.18	75.24	-0.06
136	81.28	81.05	0.23
137	80.69	80.54	0.15
138	78.53	75.93	2.60
139	79.14	77.12	2.02
140	65.00	66.97	-1.97
141	62.43	64.33	-1.90
142	73.65	73.27	0.38
143	75.09	74.93	0.16
144	63.65	66.05	-2.40
145	81.49	81.01	0.48
146	77.65	78.34	-0.69
147	77.04	76.69	0.35
148	80.91	81.03	-0.12
149	79.96	78.64	1.32
150	80.54	79.28	1.26
151	77.37	77.44	-0.07
152	78.50	78.65	-0.15
153	69.46	70.45	-0.99
154	66.37	67.96	-1.59
155	77.52	77.82	-0.30
156	76.39	75.97	0.42
157	65.23	65.39	-0.16
158	68.32	68.53	-0.21
159	76.47	75.90	0.57
160	73.47	73.60	-0.13
161	85.51	86.02	-0.51
162	79.69	80.86	-1.17
163	75.20	75.69	-0.49
164	82.94	82.98	-0.04
165	79.89	78.70	1.19
166	82.66	83.92	-1.26
167	77.07	76.58	0.49
168	80.15	83.29	-3.14
169	73.55	74.10	-0.55
170	66.68	66.07	0.61
171	76.99	77.39	-0.40
172	69.67	68.50	1.17
173	55.80	54.64	1.16
174	64.22	64.83	-0.61

For Y5: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
175	76.60	76.20	0.40
176	72.01	68.87	3.14
177	82.68	82.15	0.53
178	75.67	76.48	-0.81
179	70.73	69.94	0.79
180	79.73	79.76	-0.03
181	78.68	77.75	0.93
182	81.82	81.79	0.03
183	80.75	81.67	-0.92
184	81.84	81.88	-0.04
185	74.58	74.76	-0.18
186	66.38	67.16	-0.78
187	78.74	78.43	0.31
188	77.51	76.58	0.93
189	62.57	63.22	-0.65
190	72.61	72.46	0.15
191	74.48	74.04	0.44
192	65.14	65.85	-0.71
193	75.24	75.24	0.00

Appendix P: The CODEDBTS.IN File

The CODEDBTS.IN file is created by the RGRSSION subroutine. It contains the parameter estimates to the postulated model (eqn 2.3). The first line lists the number of design variables, the number of equation parameters, and the number of objectives (dependent variables) processed by the RGRSSION subroutine. The lines that follow first list the names of all parameters and then by objective the parameter estimations. The parameter estimations are listed in such manner as to correspondingly match the parameter names as listed (i.e., for the first objective, the INTERCEPT is matched with 86.45, the MMII*MMII is matched with -0.01609, MMIII*MMIII is matched with -0.51133, and so on).

```
12  91 5
INTERCEPT
MMII     * MMII
MMIII    * MMIII
PKEEP    * PKEEP
MICBM    * MICBM
POSC3    * POSC3
POSTRIC4 * POSTRIC4
TRID5    * TRID5
B-52PEN  * B-52PEN
B-52CMC  * B-52CMC
FB111PEN * FB111PEN
B1B      * B1B
ATB      * ATB
MMII
MMIII
PKEEP
MICBM
POSC3
POSTRIC4
TRID5
B-52PEN
B-52CMC
FB111PEN
B1B
ATB
```

```
MMII      * MMIII
MMII      * PKEEP
MMII      * MICBM
MMII      * POSC3
MMII      * POSTRIC4
MMII      * TRID5
MMII      * B-52PEN
MMII      * B-52CMC
MMII      * FB111PEN
MMII      * B1B
MMII      * ATB
MMIII     * PKEEP
MMIII     * MICBM
MMIII     * POSC3
MMIII     * POSTRIC4
MMIII     * TRID5
MMIII     * B-52PEN
MMIII     * B-52CMC
MMIII     * FB111PEN
MMIII     * B1B
MMIII     * ATB
PKEEP     * MICBM
PKEEP     * POSC3
PKEEP     * POSTRIC4
PKEEP     * TRID5
PKEEP     * B-52PEN
PKEEP     * B-52CMC
PKEEP     * FB111PEN
PKEEP     * B1B
PKEEP     * ATB
MICBM     * POSC3
MICBM     * POSTRIC4
MICBM     * TRID5
MICBM     * B-52PEN
MICBM     * B-52CMC
MICBM     * FB111PEN
MICBM     * B1B
MICBM     * ATB
POSC3     * POSTRIC4
POSC3     * TRID5
POSC3     * B-52PEN
POSC3     * B-52CMC
POSC3     * FB111PEN
POSC3     * B1B
POSC3     * ATB
POSTRIC4  * TRID5
POSTRIC4  * B-52PEN
POSTRIC4  * B-52CMC
POSTRIC4  * FB111PEN
POSTRIC4  * B1B
POSTRIC4  * ATB
TRID5     * B-52PEN
TRID5     * B-52CMC
```

TRID5 * FB111PEN
TRID5 * B1B
TRID5 * ATB
B-52PEN * B-52CMC
B-52PEN * FB111PEN
B-52PEN * B1B
B-52PEN * ATB
B-52CMC * FB111PEN
B-52CMC * B1B
B-52CMC * ATB
FB111PEN * B1B
FB111PEN * ATB
B1B * ATB

1	2	3	4	5
86.45000	88.76000	75.51000	64.99000	75.24000
-0.01609	-1.75246	0.28602	1.33445	0.25281
-0.51133	-2.34020	0.54617	2.08109	0.44516
-0.97477	-3.40793	0.82180	2.16828	0.26227
0.01656	1.54520	-0.22273	-0.92820	-0.11016
0.06102	0.18512	-0.26258	-1.55000	-0.67047
-0.02727	-0.86387	0.10586	-0.29578	-0.28852
-0.37609	-13.78309	2.20289	4.40445	-0.62625
0.34648	-0.51926	0.31742	-0.80047	-0.33078
0.11680	0.45707	0.59930	-2.11391	-0.63336
-0.00406	-1.37105	0.34977	0.29055	-0.08922
-0.12992	-3.25957	0.82430	1.07156	-0.06766
-0.07195	-9.45887	0.97586	4.65266	0.15023
0.32438	1.07297	0.20125	0.96313	0.72563
0.69203	2.44594	0.73750	4.39344	2.64250
-1.02516	7.15078	1.42563	5.86266	4.23969
-0.34062	0.61109	0.12234	2.26234	1.07547
0.38688	0.00703	0.30141	7.73688	3.38563
0.36109	0.00359	0.76156	6.35063	2.90813
-1.48375	15.93984	2.68406	9.44625	7.89703
-0.54375	-0.00109	0.45719	4.61391	1.99141
0.16781	-0.00922	-0.13781	7.62406	3.20156
0.03375	0.00078	0.06984	1.80328	0.78188
0.41266	3.76719	0.88672	5.29203	3.31078
0.06219	10.70797	0.43250	5.53359	4.75719
-0.72375	-0.70812	-0.24750	0.94375	0.10000
-0.18937	-0.96062	0.01938	0.26438	-0.11375
-0.25500	-0.62000	0.20375	0.36188	0.03687
-0.13750	0.00188	0.05375	-0.48875	-0.20875
-0.05000	0.00000	0.20375	-0.72937	-0.26313
-0.13031	-1.34125	0.13406	-0.20062	-0.35719
-0.08313	-0.00563	0.12813	-0.11000	-0.02688
-0.01188	0.00187	-0.05688	0.00063	-0.01500
0.01563	-0.00187	-0.03938	0.00312	-0.00625
0.06313	-0.00562	-0.11312	-0.01625	-0.02937
0.20250	-0.00250	0.31250	-1.09687	-0.36312
-0.73187	-2.01875	1.44000	-0.99062	-0.59187
-0.41500	-1.43687	0.32625	0.97750	0.13188

0.11625	-0.01313	0.06375	-1.11250	-0.44250
-0.34562	-0.01000	0.30125	-1.24063	-0.49563
-0.56875	-3.84438	0.75125	-0.92125	-1.10375
0.01094	0.00750	0.06188	-1.01937	-0.41219
-0.23250	-0.00125	0.98750	-2.64937	-0.90500
0.06500	0.00437	-0.10125	-0.45750	-0.21063
0.21250	-0.00562	0.73125	-3.75875	-1.38312
-0.31000	0.00125	0.14500	-2.26188	-0.95500
-0.93250	0.43375	-0.16125	0.08563	-0.02250
-0.07562	0.02375	0.18938	-2.83000	-1.15187
-0.28562	-0.00500	0.74750	-2.83187	-1.04812
-0.91125	-6.73125	1.12000	1.77313	-0.54250
0.16437	0.00250	0.68625	-2.12563	-0.71063
0.00594	-0.00031	0.65656	-2.94937	-1.08437
0.14437	0.00687	-0.17188	-0.79938	-0.36250
-0.04063	-0.02625	0.06437	-1.70875	-0.71438
0.24063	-0.02625	0.44063	-2.69875	-1.00812
0.09000	0.00000	0.22563	-1.73187	-0.66562
-0.11375	0.00000	0.01500	-0.94812	-0.41063
-0.86375	-1.16875	-0.04250	0.53813	-0.13750
-0.06500	0.00000	0.27562	-1.11188	-0.40938
-0.01000	0.00375	-0.14000	0.07812	-0.00250
-0.01125	0.00156	0.21031	-0.50531	-0.16312
-0.08125	0.00312	-0.39500	-0.17375	-0.17938
-0.02125	-0.00250	0.68375	-1.62063	-0.52063
-0.11562	-0.00625	-0.46437	-1.85187	-0.90875
-0.21000	-0.00438	-0.08750	-3.54875	-1.54500
0.63000	0.00000	0.22937	-2.59062	-0.96187
0.18437	0.00750	0.60813	-2.86937	-1.03750
0.08375	0.00000	0.21187	-1.14688	-0.42188
0.02563	-0.01000	0.17000	-1.38594	-0.54281
-0.25438	-0.02125	-0.08938	-1.37375	-0.63688
-0.22937	0.00188	0.09563	-2.31437	-0.98000
0.22812	-0.00125	0.28000	-1.85562	-0.68937
0.18562	0.00500	0.19438	-2.22563	-0.86875
0.20688	0.00062	-0.01438	-0.21688	-0.06875
0.28312	0.00625	0.19437	-1.84688	-0.69625
0.21219	0.00781	0.16719	-2.62000	-1.03781
0.26187	-0.00437	-0.14312	-2.26375	-0.95937
-0.14625	-0.00375	1.78625	-6.73188	-2.42500
-0.09313	0.00438	0.21938	-0.90562	-0.34000
-0.07687	-0.00438	0.21062	-3.49750	-1.43438
0.04063	-0.00062	-0.38563	-2.66687	-1.21375
0.13500	0.00500	0.32500	-1.71812	-0.63000
-0.05375	0.00000	0.26188	-0.55438	-0.17562
0.38063	-0.00187	0.14688	-1.37625	-0.49937
0.16750	0.00000	0.15000	-1.70563	-0.66250
-0.09313	-0.00562	0.02438	-0.38562	-0.16875
-0.07937	-0.01000	1.60938	-3.74938	-1.20250
-0.42000	0.00125	1.05875	-3.37312	-1.21625
-0.07625	-0.00062	0.37500	-1.18125	-0.41688
-0.01312	-0.00437	0.19688	-0.69437	-0.24750
0.03563	-3.96875	0.58813	0.42500	-0.52938

Appendix Q: SAS Input Code

The following code was used to execute the SAS's regression package. The same statistics as those calculated by the RGRSSION subroutine for the fifth objective were computed by the SAS regression package (refer to Appendices O and P). The purpose was to verify the RGRSSION subroutine. The SAS output for this code is shown in Appendix R.

```
options linesize=80;
filename reginput 'storage.dat';
data rsmedaem;
  infile reginput;
  input mmii mmiii pkeep micbm posc3 postric4 trid5 b52pen a b52cmc
        fb111pen b b1b c d atb y1 y2 y3 y4 y5;
  if mmii=0 then x1=-1;
  if mmii=225 then x1=0;
  if mmii=450 then x1=1;
  if mmiii=100 then x2=-1;
  if mmiii=325 then x2=0;
  if mmiii=550 then x2=1;
  if pkeep=0 then x3=-1;
  if pkeep=50 then x3=0;
  if pkeep=100 then x3=1;
  if micbm=0 then x4=-1;
  if micbm=100 then x4=0;
  if micbm=200 then x4=1;
  if posc3=176 then x5=-1;
  if posc3=240 then x5=0;
  if posc3=304 then x5=1;
  if postric4=240 then x6=-1;
  if postric4=312 then x6=0;
  if postric4=384 then x6=1;
  if trid5=0 then x7=-1;
  if trid5=168 then x7=0;
  if trid5=336 then x7=1;
  if b52pen=0 then x8=-1;
  if b52pen=78 then x8=0;
  if b52pen=156 then x8=1;
  if b52cmc=40 then x9=-1;
  if b52cmc=98 then x9=0;
  if b52cmc=156 then x9=1;
  if fb111pen=0 then xa=-1;
```

```
if fb111pen=30 then xa=0;
if fb111pen=60 then xa=1;
if b1b=30 then xb=-1;
if b1b=65 then xb=0;
if b1b=100 then xb=1;
if atb=0 then xc=-1;
if atb=62 then xc=0;
if atb=124 then xc=1;
x11=x1*x1;
x12=x1*x2;
x13=x1*x3;
x14=x1*x4;
x15=x1*x5;
x16=x1*x6;
x17=x1*x7;
x18=x1*x8;
x19=x1*x9;
x1a=x1*xa;
x1b=x1*xb;
x1c=x1*xc;
x22=x2*x2;
x23=x2*x3;
x24=x2*x4;
x25=x2*x5;
x26=x2*x6;
x27=x2*x7;
x28=x2*x8;
x29=x2*x9;
x2a=x2*xa;
x2b=x2*xb;
x2c=x2*xc;
x33=x3*x3;
x34=x3*x4;
x35=x3*x5;
x36=x3*x6;
x37=x3*x7;
x38=x3*x8;
x39=x3*x9;
x3a=x3*xa;
x3b=x3*xb;
x3c=x3*xc;
x44=x4*x4;
x45=x4*x5;
x46=x4*x6;
x47=x4*x7;
x48=x4*x8;
x49=x4*x9;
x4a=x4*xa;
x4b=x4*xb;
x4c=x4*xc;
x55=x5*x5;
x56=x5*x6;
x57=x5*x7;
```

```
  x58=x5*x8;
  x59=x5*x9;
  x5a=x5*xa;
  x5b=x5*xb;
  x5c=x5*xc;
  x66=x6*x6;
  x67=x6*x7;
  x68=x6*x8;
  x69=x6*x9;
  x6a=x6*xa;
  x6b=x6*xb;
  x6c=x6*xc;
  x77=x7*x7;
  x78=x7*x8;
  x79=x7*x9;
  x7a=x7*xa;
  x7b=x7*xb;
  x7c=x7*xc;
  x88=x8*x8;
  x89=x8*x9;
  x8a=x8*xa;
  x8b=x8*xb;
  x8c=x8*xc;
  x99=x9*x9;
  x9a=x9*xa;
  x9b=x9*xb;
  x9c=x9*xc;
  xaa=xa*xa;
  xab=xa*xb;
  xac=xa*xc;
  xbb=xb*xb;
  xbc=xb*xc;
  xcc=xc*xc;
proc reg;
  model y5 = x1 x2 x3 x4 x5 x6 x7 x8 x9 xa xb xc x11 x12 x13 x14 x15 x16
             x17 x18 x19 x1a x1b x1c x22 x23 x24 x25 x26 x27 x28 x29 x2a
             x2b x2c x33 x34 x35 x36 x37 x38 x39 x3a x3b x3c x44 x45 x46
             x47 x48 x49 x4a x4b x4c x55 x56 x57 x58 x59 x5a x5b x5c x66
             x67 x68 x69 x6a x6b x6c x77 x78 x79 x7a x7b x7c x88 x89 x8a
             x8b x8c x99 x9a x9b x9c xaa xab xac xbb xbc xcc / ss1 p;
```

Appendix R: SAS Output For The Fifth Objective

The regression output from SAS for the fifth objective is listed below. To verify the RGRSSION subroutine, the statistics that the RGRSSION subroutine produces (Appendices O and P) was compared to the same statistic produced by SAS. [Note: SAS does not list the parameter estimates in the same order as Appendix P does.]

SAS 17:18 FRIDAY, MARCH 13, 1987 33

DEP VARIABLE: Y5

ANALYSIS OF VARIANCE

SOURCE	DF	SUM OF SQUARES	MEAN SQUARE	F VALUE	PROB>F
MODEL	90	10786.68	119.852	45.642	0.0001
ERROR	102	267.8438	2.62592		
C TOTAL	192	11054.52			

ROOT MSE	1.620469	R-SQUARE	0.9758
DEP MEAN	74.6743	ADJ R-SQ	0.9544
C.V.	2.170049		

PARAMETER ESTIMATES

VARIABLE	DF	PARAMETER ESTIMATE	STANDARD ERROR	T FOR H0: PARAMETER=0	PROB > |T|
INTERCEP	1	75.24	1.620469	46.431	0.0001
X1	1	0.725625	0.2025586	3.582	0.0005
X2	1	2.6425	0.2025586	13.046	0.0001
X3	1	4.239688	0.2025586	20.931	0.0001
X4	1	1.075469	0.2025586	5.309	0.0001
X5	1	3.385625	0.2025586	16.714	0.0001
X6	1	2.908125	0.2025586	14.357	0.0001
X7	1	7.897031	0.2025586	38.986	0.0001
X8	1	1.991406	0.2025586	9.831	0.0001
X9	1	3.201563	0.2025586	15.806	0.0001
XA	1	0.781875	0.2025586	3.860	0.0002
XB	1	3.310781	0.2025586	16.345	0.0001
XC	1	4.757188	0.2025586	23.485	0.0001
X11	1	0.2528125	0.4723353	0.535	0.5936
X12	1	0.1	0.4051173	0.247	0.8055

X13	1	-0.11375	0.4051173	-0.281	0.7794
X14	1	0.036875	0.4051173	0.091	0.9277
X15	1	-0.20875	0.4051173	-0.515	0.6075
X16	1	-0.263125	0.4051173	-0.650	0.5175
X17	1	-0.357188	0.2864612	-1.247	0.2153
X18	1	-0.026875	0.4051173	-0.066	0.9472
X19	1	-0.015	0.4051173	-0.037	0.9705
X1A	1	-0.00625	0.4051173	-0.015	0.9877
X1B	1	-0.029375	0.4051173	-0.073	0.9423
X1C	1	-0.363125	0.4051173	-0.896	0.3722
X22	1	0.4451562	0.4723353	0.942	0.3482
X23	1	-0.591875	0.4051173	-1.461	0.1471
X24	1	0.131875	0.4051173	0.326	0.7455
X25	1	-0.4425	0.4051173	-1.092	0.2773
X26	1	-0.495625	0.4051173	-1.223	0.2240
X27	1	-1.10375	0.4051173	-2.725	0.0076
X28	1	-0.412187	0.2864612	-1.439	0.1532
X29	1	-0.905	0.4051173	-2.234	0.0277
X2A	1	-0.210625	0.4051173	-0.520	0.6043
X2B	1	-1.38313	0.4051173	-3.414	0.0009
X2C	1	-0.955	0.4051173	-2.357	0.0203
X33	1	0.2622656	0.4723353	0.555	0.5799
X34	1	-0.0225	0.4051173	-0.056	0.9558
X36	1	-1.04813	0.4051173	-2.587	0.0111
X37	1	-0.5425	0.4051173	-1.339	0.1835
X38	1	-0.710625	0.4051173	-1.754	0.0824
X39	1	-1.08438	0.2864612	-3.785	0.0003
X3A	1	-0.3625	0.4051173	-0.895	0.3730
X3B	1	-0.714375	0.4051173	-1.763	0.0808
X3C	1	-1.00812	0.4051173	-2.488	0.0144
X44	1	-0.110156	0.4723353	-0.233	0.8161
X45	1	-0.665625	0.4051173	-1.643	0.1035
X46	1	-0.410625	0.4051173	-1.014	0.3132
X47	1	-0.1375	0.4051173	-0.339	0.7350
X48	1	-0.409375	0.4051173	-1.011	0.3146
X49	1	-0.0025	0.4051173	-0.006	0.9951
X4A	1	-0.163125	0.2864612	-0.569	0.5703
X4B	1	-0.179375	0.4051173	-0.443	0.6589
X4C	1	-0.520625	0.4051173	-1.285	0.2017
X55	1	-0.670469	0.4723353	-1.419	0.1588
X56	1	-0.90875	0.4051173	-2.243	0.0270
X57	1	-1.545	0.4051173	-3.814	0.0002
X58	1	-0.961875	0.4051173	-2.374	0.0195
X59	1	-1.0375	0.4051173	-2.561	0.0119
X5A	1	-0.421875	0.4051173	-1.041	0.3002
X5B	1	-0.542812	0.2864612	-1.895	0.0609
X5C	1	-0.636875	0.4051173	-1.572	0.1190
X66	1	-0.288516	0.4723353	-0.611	0.5427
X67	1	-0.98	0.4051173	-2.419	0.0173
X68	1	-0.689375	0.4051173	-1.702	0.0919
X69	1	-0.86875	0.4051173	-2.144	0.0344
X6A	1	-0.06875	0.4051173	-0.170	0.8656
X6B	1	-0.69625	0.4051173	-1.719	0.0887

VARIABLE	DF	PARAMETER ESTIMATE	STANDARD ERROR	T FOR H0: PARAMETER=0	PROB > |T|
X6C	1	-1.03781	0.2864612	-3.623	0.0005
X77	1	-0.62625	0.4723353	-1.326	0.1878
X78	1	-0.959375	0.4051173	-2.368	0.0198
X79	1	-2.425	0.4051173	-5.986	0.0001
X7A	1	-0.34	0.4051173	-0.839	0.4033
X7B	1	-1.43438	0.4051173	-3.541	0.0006
X7C	1	-1.21375	0.4051173	-2.996	0.0034
X88	1	-0.330781	0.4723353	-0.700	0.4853
X89	1	-0.63	0.4051173	-1.555	0.1230
X8A	1	-0.175625	0.4051173	-0.434	0.6656
X8B	1	-0.499375	0.4051173	-1.233	0.2205
X8C	1	-0.6625	0.4051173	-1.635	0.1051
X99	1	-0.633359	0.4723353	-1.341	0.1829
X9A	1	-0.16875	0.4051173	-0.417	0.6779
X9B	1	-1.2025	0.4051173	-2.968	0.0037
X9C	1	-1.21625	0.4051173	-3.002	0.0034
XAA	1	-0.0892188	0.4723353	-0.189	0.8506
XAB	1	-0.416875	0.4051173	-1.029	0.3059
XAC	1	-0.2475	0.4051173	-0.611	0.5426
XBB	1	-0.0676563	0.4723353	-0.143	0.8864
XBC	1	-0.529375	0.4051173	-1.307	0.1942

VARIABLE	DF	TYPE I SS
INTERCEP	1	1076216
X1	1	33.69803
X2	1	446.8996
X3	1	1150.397
X4	1	74.02451
X5	1	733.5972
X6	1	541.2602
X7	1	3991.239
X8	1	253.8047
X9	1	656.0002
XA	1	39.12503
XB	1	701.5214
XC	1	1448.373
X11	1	2.841361
X12	1	0.16
X13	1	0.207025
X14	1	0.02175625
X15	1	0.697225
X16	1	1.107756
X17	1	4.082653
X18	1	0.01155625
X19	1	0.0036
X1A	1	0.000625
X1B	1	0.01380625
X1C	1	2.109756

VARIABLE	DF	TYPE I SS
X22	1	16.45276
X23	1	5.605056
X24	1	0.2782562
X25	1	3.1329
X26	1	3.930306
X27	1	19.49223
X28	1	5.436753
X29	1	13.1044
X2A	1	0.7098062
X2B	1	30.60856
X2C	1	14.5924
X33	1	6.486849
X34	1	0.0081
X35	1	21.22906
X36	1	17.57706
X37	1	4.7089
X38	1	8.079806
X39	1	37.62781
X3A	1	2.1025
X3B	1	8.165306
X3C	1	16.26106
X44	1	2.409523
X45	1	7.088906
X46	1	2.697806
X47	1	0.3025
X48	1	2.681406
X49	1	0.0001
X4A	1	0.8515125
X4B	1	0.5148063
X55	1	5.656306
X56	1	13.21322
X57	1	38.1924
X58	1	14.80326
X59	1	17.2225
X5A	1	2.847656
X5B	1	9.428653
X5C	1	6.489756
X66	1	0.3317038
X67	1	15.3664
X68	1	7.603806
X69	1	12.07563
X6A	1	0.075625
X6B	1	7.756225
X6C	1	34.46575
X77	1	7.151862
X78	1	14.72641
X79	1	94.09
X7A	1	1.8496
X7B	1	32.91891
X7C	1	23.57103
X88	1	1.184964

VARIABLE	DF	TYPE I SS
X89	1	6.3504
X8A	1	0.4935063
X8B	1	3.990006
X8C	1	7.0225
X99	1	13.12689
X9A	1	0.455625
X9B	1	23.1361
X9C	1	23.66823
XAA	1	0.4612686
XAB	1	2.780556
XAC	1	0.9801
XBB	1	0.7704485
XBC	1	4.483806
XCC	1	0.2656558

OBS	ACTUAL	PREDICT VALUE	RESIDUAL
1	85.75	85.7348	.0151563
2	74.36	75.9527	-1.5927
3	68.57	67.3939	1.17609
4	82.9	83.3561	-.456094
5	80.32	79.1936	1.12641
7	82.53	83.223	-.692969
8	85.01	85.2155	-.205469
9	73.23	74.0045	-.774531
10	64.12	64.6108	-.490781
11	82.16	82.0017	0.158281
12	79.48	78.2392	1.24078
13	56.77	56.4333	0.336719
14	67.57	67.597	-.027031
15	69.47	69.1452	0.324844
16	58.67	58.8164	-.146406
17	83.88	83.162	0.717969
18	82.44	82.8036	-.363594
19	79.43	78.6961	0.733906
20	81.03	81.812	-.782031
21	73.86	73.8414	.0185938
22	78.27	79.3839	-1.1139
23	74.9	74.7308	0.169219
24	81.25	80.8764	0.373594
25	79.58	78.8692	0.710781
26	72.99	72.7792	0.210781
27	78.11	77.5439	0.566094
28	66.84	67.2058	-.365781
29	59.61	59.5986	.0114063
30	69.76	69.8811	-.121094
31	76.52	76.183	0.337031
32	66.78	67.883	-1.103

OBS	ACTUAL	PREDICT VALUE	RESIDUAL
33	86.08	86.3319	-.251875
34	84.25	86.9525	-2.7025
35	63.71	67.6684	-3.9584
36	75.38	76.7478	-1.3678
37	74.12	74.3719	-.251875
38	85.41	84.5059	0.904063
39	81.1	79.2353	1.86469
40	81.88	81.1512	0.72875
41	79.79	77.4344	2.35563
42	57.17	55.9803	1.18969
43	68.83	69.3972	-.567187
44	67.53	66.9312	0.59875
45	55.86	53.5044	2.35563
46	78.97	75.5084	3.46156
47	83.59	85.1166	-1.5266
48	62.45	65.2825	-2.8325
49	80.8	78.4763	2.32375
50	80.16	78.4338	1.72625
51	78.33	76.8422	1.48781
52	79.09	77.5872	1.50281
53	70.63	71.0672	-.437188
55	72.33	73.4666	-1.1366
56	79.6	78.8016	0.798437
57	78.84	78.1066	0.733437
58	72.92	74.8775	-1.9575
59	75.95	76.275	-0.325
60	65.07	67.0925	-2.0225
61	61.98	64.0075	-2.0275
62	69.23	71.0841	-1.8541
63	74.75	74.0738	0.67625
64	67.54	68.6347	-1.0947
65	83.37	81.1295	2.24047
66	79.47	79.3911	.0789063
67	75.37	74.3955	0.974531
68	81.13	80.9439	0.186094
69	76.76	76.6002	0.159844
70	80.34	80.2608	.0792188
71	75.53	74.8327	0.697344
72	80.34	79.3364	1.00359
73	76.29	75.4267	0.863281
74	64.53	66.2811	-1.7511
75	76.02	75.0008	1.01922
76	65.92	67.022	-1.102
77	54.3	55.5173	-1.2173
78	66.28	68.138	-1.858
79	76.39	75.7373	0.652656
80	65.24	67.2667	-2.0267
81	86.48	86.8427	-.362656
82	79.88	82.3264	-2.4464
83	79.38	81.0852	-1.7052

OBS	ACTUAL	PREDICT VALUE	RESIDUAL
84	85.98	86.5914	-.611406
85	74.4	74.5048	-.104844
86	75	76.1161	-1.1161
87	71.84	70.553	1.28703
88	83.84	85.1995	-1.3595
89	77.09	76.532	0.557969
90	76.55	75.0158	1.53422
91	83.35	84.6733	-1.3233
92	70.66	68.6667	1.99328
93	54.97	54.1542	0.815781
94	58.06	57.0305	1.02953
95	67.65	66.7448	0.905156
96	65.05	64.1436	0.906406
97	85.18	85.0864	.0935938
98	81.68	82.3911	-.711094
99	80.2	79.382	0.817969
100	83.76	84.0748	-.314844
101	70.65	69.9452	0.704844
103	72.04	72.7411	-.701094
104	84.45	84.462	-.012031
105	80.93	81.6492	-.719219
104	84.45	84.462	-.012031
105	80.93	81.6492	-.719219
106	79.38	78.5327	0.847344
107	83.03	83.343	-.312969
108	68.51	67.7845	0.725469
109	57.26	57.2367	.0232813
110	64.52	64.1908	0.329219
111	66.66	66.3614	0.298594
112	59.41	59.5148	-.104844
113	83.45	82.683	0.767031
114	76.56	78.8361	-2.2761
115	74.34	73.0705	1.26953
116	81.5	81.7823	-.282344
117	79.87	78.6889	1.18109
118	81.96	82.1095	-.149531
119	78.96	79.7202	-.760156
120	80.57	81.9423	-1.3723
121	73.43	74.2755	-.845469
122	64.53	64.8898	-.359844
123	78.45	77.4217	1.02828
124	74.62	72.6795	1.94047
125	57.24	57.4977	-.257656
126	69.35	69.4033	-.053281
127	74.92	75.6127	-.692656
128	68.19	67.327	0.862969
129	82.11	82.2306	-.120625
130	81.54	81.7419	-.201875
131	79.43	77.2	2.23
132	80.01	78.3637	1.64625

OBS	ACTUAL	PREDICT VALUE	RESIDUAL
133	66.49	68.6681	-2.1781
134	76.3	76.8725	-0.5725
135	75.18	75.2362	-0.05625
136	81.28	81.0494	0.230625
137	80.69	80.5356	0.154375
138	78.53	75.9337	2.59625
139	79.14	77.1225	2.0175
140	65	66.9719	-1.9719
141	62.43	64.3331	-1.9031
142	73.65	73.2725	0.3775
143	75.09	74.9337	0.15625
144	63.65	66.0544	-2.4044
145	81.49	81.0075	0.4825
146	77.65	78.3447	-.694688
147	77.04	76.6947	0.345312
148	80.91	81.025	-0.115
149	79.96	78.6428	1.31719
151	77.37	77.4441	-.074063
152	78.5	78.6463	-0.14625
153	69.46	70.4509	-.990938
154	66.37	67.9584	-1.5884
155	77.52	77.8213	-0.30125
156	76.39	75.9666	0.423437
157	65.23	65.3863	-0.15625
158	68.32	68.5313	-0.21125
159	76.47	75.8975	0.5725
160	73.47	73.595	-0.125
161	85.51	86.0242	-.514219
162	79.69	80.8586	-1.1686
163	75.2	75.6927	-.492656
164	82.94	82.9758	-.035781
165	79.89	78.6964	1.19359
166	82.66	83.9161	-1.2561
167	77.07	76.578	0.492031
168	80.15	83.2936	-3.1436
169	73.55	74.0955	-.545469
170	66.68	66.072	0.607969
171	76.99	77.3877	-.397656
172	69.67	68.5008	1.16922
173	55.8	54.6377	1.16234
174	64.22	64.8323	-.612344
175	76.6	76.203	0.397031
176	72.01	68.8658	3.14422
177	82.68	82.1523	0.527656
178	75.67	76.4811	-.811094
179	70.73	69.9367	0.793281
180	79.73	79.7592	-.029219
181	78.68	77.7545	0.925469
182	81.82	81.7902	.0298437
183	80.75	81.6652	-.915156

OBS	ACTUAL	PREDICT VALUE	RESIDUAL
184	81.84	81.8798	-.039844
185	74.58	74.7561	-.176094
186	66.38	67.1592	-.779219
187	78.74	78.4342	0.305781
188	77.51	76.577	0.932969
189	62.57	63.2195	-.649531
190	72.61	72.4589	0.151094
191	74.48	74.0364	0.443594
192	65.14	65.8495	-.709531
193	75.24	75.24	-4.1E-13

SUM OF RESIDUALS -2.66454E-14
SUM OF SQUARED RESIDUALS 267.8438

Appendix S: The XCUTEQNS.IN File

The XCUTEQNS.IN File is created by the DECODE subroutine. It contains the decoded parameter estimates for the postulated equation (equation 2.3 in this case). The first line shows the number of design variables, the number of parameter estimates, and the number of objectives processed by the DECODE subroutine. The next number of lines show the name of each design parameter along with its range of values. Last, the remaining lines first give which objective and then the parameter estimates for that objective. The values shown in this appendix are the decoded parameter estimates for the coded parameter estimates shown, by objective, in Appendix P.

```
12  91 5
MMII          0.00   450.00
MMIII       100.00   550.00
PKEEP         0.00   100.00
MICBM         0.00   200.00
POSC3       176.00   304.00
POSTRIC4    240.00   384.00
TRID5         0.00   336.00
B-52PEN       0.00   156.00
B-52CMC      40.00   156.00
FB111PEN      0.00    60.00
B1B          30.00   100.00
ATB           0.00   124.00
```

1	2	3	4	5
77.324264176122	-44.068251999832	108.617774456922	-358.700617186886	-125.214476892050
0.011007911148	0.037923677723	-0.006888984802	0.012140779900	0.012928076306
0.026233718634	0.076918390324	-0.034608905814	0.102736746640	0.054679718141
0.101039127696	0.482789803640	-0.188186919048	0.749967267105	0.387007125205
0.026976605200	0.009399783963	-0.010141193473	0.157526907649	0.069343867570
-0.003484952535	-0.020772220255	0.033040349702	0.783224201218	0.333503145141
0.010032719584	0.104225414709	-0.014357538219	0.504962326655	0.233209622389
0.023554792172	0.347164417615	-0.042884634142	0.279168491762	0.185094193969
-0.076030512259	0.013263306598	-0.057963822172	0.497093737632	0.189473105676
-0.012053095167	-0.027334587273	-0.230260844540	1.052439899331	0.380679221401

-0.031744888661	0.091265280058	-0.087682051040	0.447496049808	0.183295167488
-0.022999209907	0.568970436625	-0.237347513019	0.965873900930	0.469928564274
0.011530612100	0.597974391233	-0.103181827207	0.506347899584	0.318958846779
-0.000000317827	-0.000034616494	0.000005649778	0.000026359506	0.000004993778
-0.000014296296	-0.000013987556	-0.000004888889	0.000018641975	0.000001975309
-0.000016832889	-0.000085388444	0.000001722667	0.000023500444	-0.000010111111
-0.000011333333	-0.000027555556	0.000009055556	0.000016083556	0.000001638667
-0.000009548611	0.000000130556	0.000003732639	-0.000033940972	-0.000014496528
-0.000003086420	0.000000000000	0.000012577160	-0.000045022840	-0.000016242593
-0.000003447354	-0.000035482804	0.000003546561	-0.000005307407	-0.000009449471
-0.000004736752	-0.000000320798	0.000007300855	-0.000006267806	-0.000001531624
-0.000000910345	0.000000143295	-0.000004358621	0.000000048276	-0.000001149425
0.000002315556	-0.000000277037	-0.000005834074	0.000000462222	-0.000000925926
0.000008016508	-0.000000713651	-0.000014364444	-0.000002063492	-0.000003729524
0.000014516129	-0.000000179211	0.000022401434	-0.000078628674	-0.000026030108
-0.000010100346	-0.000046226173	0.000010788543	0.000041107951	0.000008793284
-0.000065055111	-0.000179444444	0.000128000000	-0.000088055111	-0.000052610667
-0.000018444444	-0.000063860889	0.000014500000	0.000043444444	0.000005861333
0.000008072917	-0.000000911806	0.000004427083	-0.000077256944	-0.000030729167
-0.000021334568	-0.000000617284	0.000018595679	-0.000076582099	-0.000030594444
-0.000015046296	-0.000101703175	0.000019874339	-0.000024371693	-0.000029199735
0.000000623362	0.000000427350	0.000003525926	-0.000058083761	-0.000023486610
-0.000017816092	-0.000000095785	0.000075670498	-0.000203016858	-0.000069348659
0.000009629630	0.000000647407	-0.000015000000	-0.000067777778	-0.000031204444
0.000026984127	-0.000000713651	0.000092857143	-0.000477301587	-0.000175634286
-0.000022222222	0.000000089606	0.000010394265	-0.000162141935	-0.000068458781
-0.000389908000	-0.001363172000	0.000328720000	0.000867312000	0.000104908000
-0.000186500000	0.000086750000	-0.000032250000	0.000017126000	-0.000004500000
-0.000023631250	0.000007421875	0.000059181250	-0.000884375000	-0.000359959375
-0.000079338889	-0.000001388889	0.000207638889	-0.000786630556	-0.000291144444
-0.000108482143	-0.000801339286	0.000133333333	0.000211086905	-0.000064583333
0.000042146154	0.000000641026	0.000175961538	-0.000545033333	-0.000182212821
0.000002048276	-0.000000106897	0.000226400000	-0.001017024138	-0.000373920690
0.000096246667	0.000004580000	-0.000114586667	-0.000532920000	-0.000241666667
-0.000023217143	-0.000015000000	0.000036782857	-0.000976428571	-0.000408217143
0.000077622581	-0.000008467742	0.000142138710	-0.000870564516	-0.000325200000
0.000001656000	0.000154520000	-0.000022273000	-0.000092820000	-0.000011016000
0.000014062500	0.000000000000	0.000035254687	-0.000270604687	-0.000104003125
-0.000015798611	0.000000000000	0.000002083333	-0.000131683333	-0.000057031944
-0.000051413690	-0.000069568452	-0.000002529762	0.000032031548	-0.000008184524
-0.000008333333	0.000000000000	0.000035335897	-0.000142548718	-0.000052484615
-0.000001724138	0.000000646552	-0.000024137931	0.000013468966	-0.000000431034
-0.000003750000	0.000000520000	0.000070103333	-0.000168436667	-0.000054373333
-0.000023214286	0.000000891429	-0.000112857143	-0.000049642857	-0.000051251429
-0.000003427419	-0.000000403226	0.000110282258	-0.000261391935	-0.000083972581
0.000014897461	0.000045195312	-0.000064106445	-0.000378417969	-0.000163688965
-0.000025091146	-0.000001356337	-0.000100774740	-0.000401881510	-0.000197211372
-0.000019531250	-0.000000407366	-0.000008138021	-0.000330054874	-0.000143694196
0.000126201923	0.000000000000	0.000045947516	-0.000518954327	-0.000192682292
0.000049668642	0.000002020474	0.000163828125	-0.000772998384	-0.000279498922
0.000043619792	0.000000000000	0.000110348958	-0.000597333333	-0.000219729167
0.000011441964	-0.000004464286	0.000075892857	-0.000618723214	-0.000242325893
-0.000064107863	-0.000005355343	-0.000022525202	-0.000346207157	-0.000160504032

-0.000005260417	-0.000166641590	0.000020420525	-0.000057056327	-0.000055655864
-0.000018962467	0.000000155423	0.000007905919	-0.000191333499	-0.000081018519
0.000040619658	-0.000000222578	0.000049857550	-0.000330416667	-0.000122751068
0.000044449234	0.000001197318	0.000046546935	-0.000532957375	-0.000208034004
0.000095777778	0.000000287037	-0.000006657407	-0.000100407407	-0.000031828704
0.000112349206	0.000002480159	0.000077130952	-0.000732888889	-0.000276289683
0.000047533602	0.000001749552	0.000037452957	-0.000586917563	-0.000232484319
-0.000013325184	-0.000488346443	0.000078050241	0.000156053359	-0.000022188563
0.000019983974	-0.000000333486	-0.000010921856	-0.000172752595	-0.000073211996
-0.000015009236	-0.000000384852	0.000183317939	-0.000690874384	-0.000248871100
-0.000018478175	0.000000869048	0.000043527778	-0.000179686508	-0.000067460317
-0.000013073129	-0.000000744898	0.000035819728	-0.000594812925	-0.000243942177
0.000003900730	-0.000000059524	-0.000037022849	-0.000256035906	-0.000116527458
0.000056949375	-0.000085348455	0.000052172913	-0.000131569691	-0.000054368836
0.000029840849	0.000001105217	0.000071839080	-0.000379778957	-0.000139257294
-0.000022970085	0.000000000000	0.000111914530	-0.000236914530	-0.000075051282
0.000139424908	-0.000000684982	0.000053802198	-0.000504120879	-0.000182919414
0.000034636063	0.000000000000	0.000031017370	-0.000352694376	-0.000136993383
0.000034720571	0.000135870987	0.000178151011	-0.000628391795	-0.000188275862
-0.000053522989	-0.000003229885	0.000014011494	-0.000221620690	-0.000096982759
-0.000039098522	-0.000004926108	0.000792798030	-0.001846985222	-0.000592364532
-0.000116796440	0.000000347608	0.000294424360	-0.000938020022	-0.000338223026
-0.000004511111	-0.001523388889	0.000388633333	0.000322833333	-0.000099133333
-0.000072619048	-0.000000590476	0.000357142857	-0.001125000000	-0.000397028571
-0.000007053763	-0.000002349462	0.000105849462	-0.000373317204	-0.000133064516
-0.000106057143	-0.002660873469	0.000672897959	0.000874742857	-0.000055232653
0.000016419355	-0.001828917051	0.000271027650	0.000195852535	-0.000243953917
-0.000018717482	-0.002460684183	0.000253865765	0.001210369407	0.000039081686

Appendix T: The DECODE.OUT File

The DECODE.OUT file is created from the DECODE subroutine. The file contains the decoded parameter estimates along with their respective term name for each response objective processed through the DECODE subroutine.

The decoded parameter estimates for Y1 are:

INTERCEPT		77.324264176122
MMII		0.011007911148
MMIII		0.026233718634
PKEEP		0.101039127696
MICBM		0.026976605200
POSC3		-0.003484952535
POSTRIC4		0.010032719584
TRID5		0.023554792172
B-52PEN		-0.076030512259
B-52CMC		-0.012053095167
FB111PEN		-0.031744888661
B1B		-0.022999209907
ATB		0.011530612100
MMII	* MMII	-0.000000317827
MMII	* MMIII	-0.000014296296
MMII	* PKEEP	-0.000016832889
MMII	* MICBM	-0.000011333333
MMII	* POSC3	-0.000009548611
MMII	* POSTRIC4	-0.000003086420
MMII	* TRID5	-0.000003447354
MMII	* B-52PEN	-0.000004736752
MMII	* B-52CMC	-0.000000910345
MMII	* FB111PEN	0.000002315556
MMII	* B1B	0.000008016508
MMII	* ATB	0.000014516129
MMIII	* MMIII	-0.000010100346
MMIII	* PKEEP	-0.000065055111
MMIII	* MICBM	-0.000018444444
MMIII	* POSC3	0.000008072917
MMIII	* POSTRIC4	-0.000021334568
MMIII	* TRID5	-0.000015046296
MMIII	* B-52PEN	0.000000623362
MMIII	* B-52CMC	-0.000017816092
MMIII	* FB111PEN	0.000009629630
MMIII	* B1B	0.000026984127

The decoded parameter estimates for Y1 are:

MMIII	*	ATB	-0.000022222222
PKEEP	*	PKEEP	-0.000389908000
PKEEP	*	MICBM	-0.000186500000
PKEEP	*	POSC3	-0.000023631250
PKEEP	*	POSTRIC4	-0.000079338889
PKEEP	*	TRID5	-0.000108482143
PKEEP	*	B-52PEN	0.000042146154
PKEEP	*	B-52CMC	0.000002048276
PKEEP	*	FB111PEN	0.000096246667
PKEEP	*	B1B	-0.000023217143
PKEEP	*	ATB	0.000077622581
MICBM	*	MICBM	0.000001656000
MICBM	*	POSC3	0.000014062500
MICBM	*	POSTRIC4	-0.000015798611
MICBM	*	TRID5	-0.000051413690
MICBM	*	B-52PEN	-0.000008333333
MICBM	*	B-52CMC	-0.000001724138
MICBM	*	FB111PEN	-0.000003750000
MICBM	*	B1B	-0.000023214286
MICBM	*	ATB	-0.000003427419
POSC3	*	POSC3	0.000014897461
POSC3	*	POSTRIC4	-0.000025091146
POSC3	*	TRID5	-0.000019531250
POSC3	*	B-52PEN	0.000126201923
POSC3	*	B-52CMC	0.000049668642
POSC3	*	FB111PEN	0.000043619792
POSC3	*	B1B	0.000011441964
POSC3	*	ATB	-0.000064107863
POSTRIC4	*	POSTRIC4	-0.000005260417
POSTRIC4	*	TRID5	-0.000018962467
POSTRIC4	*	B-52PEN	0.000040619658
POSTRIC4	*	B-52CMC	0.000044449234
POSTRIC4	*	FB111PEN	0.000095777778
POSTRIC4	*	B1B	0.000112349206
POSTRIC4	*	ATB	0.000047533602
TRID5	*	TRID5	-0.000013325184
TRID5	*	B-52PEN	0.000019983974
TRID5	*	B-52CMC	-0.000015009236
TRID5	*	FB111PEN	-0.000018478175
TRID5	*	B1B	-0.000013073129
TRID5	*	ATB	0.000003900730
B-52PEN	*	B-52PEN	0.000056949375
B-52PEN	*	B-52CMC	0.000029840849
B-52PEN	*	FB111PEN	-0.000022970085
B-52PEN	*	B1B	0.000139424908
B-52PEN	*	ATB	0.000034636063
B-52CMC	*	B-52CMC	0.000034720571
B-52CMC	*	FB111PEN	-0.000053522989
B-52CMC	*	B1B	-0.000039098522
B-52CMC	*	ATB	-0.000116796440

The decoded parameter estimates for Y1 are:

FB111PEN	*	FB111PEN	-0.000004511111
FB111PEN	*	B1B	-0.000072619048
FB111PEN	*	ATB	-0.000007053763
B1B	*	B1B	-0.000106057143
B1B	*	ATB	0.000016419355
ATB	*	ATB	-0.000018717482

The decoded parameter estimates for Y2 are:

INTERCEPT			-44.068251999832
MMII			0.037923677723
MMIII			0.076918390324
PKEEP			0.482789803640
MICBM			0.009399783963
POSC3			-0.020772220255
POSTRIC4			0.104225414709
TRID5			0.347164417615
B-52PEN			0.013263306598
B-52CMC			-0.027334587273
FB111PEN			0.091265280058
B1B			0.568970436625
ATB			0.597974391233
MMII	*	MMII	-0.000034616494
MMII	*	MMIII	-0.000013987556
MMII	*	PKEEP	-0.000085388444
MMII	*	MICBM	-0.000027555556
MMII	*	POSC3	0.000000130556
MMII	*	POSTRIC4	0.000000000000
MMII	*	TRID5	-0.000035482804
MMII	*	B-52PEN	-0.000000320798
MMII	*	B-52CMC	0.000000143295
MMII	*	FB111PEN	-0.000000277037
MMII	*	B1B	-0.000000713651
MMII	*	ATB	-0.000000179211
MMIII	*	MMIII	-0.000046226173
MMIII	*	PKEEP	-0.000179444444
MMIII	*	MICBM	-0.000063860889
MMIII	*	POSC3	-0.000000911806
MMIII	*	POSTRIC4	-0.000000617284
MMIII	*	TRID5	-0.000101703175
MMIII	*	B-52PEN	0.000000427350
MMIII	*	B-52CMC	-0.000000095785
MMIII	*	FB111PEN	0.000000647407
MMIII	*	B1B	-0.000000713651
MMIII	*	ATB	0.000000089606
PKEEP	*	PKEEP	-0.001363172000
PKEEP	*	MICBM	0.000086750000

The decoded parameter estimates for Y2 are:

PKEEP	*	POSC3	0.000007421875
PKEEP	*	POSTRIC4	-0.000001388889
PKEEP	*	TRID5	-0.000801339286
PKEEP	*	B-52PEN	0.000000641026
PKEEP	*	B-52CMC	-0.000000106897
PKEEP	*	FB111PEN	0.000004580000
PKEEP	*	B1B	-0.000015000000
PKEEP	*	ATB	-0.000008467742
MICBM	*	MICBM	0.000154520000
MICBM	*	POSC3	0.000000000000
MICBM	*	POSTRIC4	0.000000000000
MICBM	*	TRID5	-0.000069568452
MICBM	*	B-52PEN	0.000000000000
MICBM	*	B-52CMC	0.000000646552
MICBM	*	FB111PEN	0.000000520000
MICBM	*	B1B	0.000000891429
MICBM	*	ATB	-0.000000403226
POSC3	*	POSC3	0.000045195312
POSC3	*	POSTRIC4	-0.000001356337
POSC3	*	TRID5	-0.000000407366
POSC3	*	B-52PEN	0.000000000000
POSC3	*	B-52CMC	0.000002020474
POSC3	*	FB111PEN	0.000000000000
POSC3	*	B1B	-0.000004464286
POSC3	*	ATB	-0.000005355343
POSTRIC4	*	POSTRIC4	-0.000166641590
POSTRIC4	*	TRID5	0.000000155423
POSTRIC4	*	B-52PEN	-0.000000222578
POSTRIC4	*	B-52CMC	0.000001197318
POSTRIC4	*	FB111PEN	0.000000287037
POSTRIC4	*	B1B	0.000002480159
POSTRIC4	*	ATB	0.000001749552
TRID5	*	TRID5	-0.000488346443
TRID5	*	B-52PEN	-0.000000333486
TRID5	*	B-52CMC	-0.000000384852
TRID5	*	FB111PEN	0.000000869048
TRID5	*	B1B	-0.000000744898
TRID5	*	ATB	-0.000000059524
B-52PEN	*	B-52PEN	-0.000085348455
B-52PEN	*	B-52CMC	0.000001105217
B-52PEN	*	FB111PEN	0.000000000000
B-52PEN	*	B1B	-0.000000684982
B-52PEN	*	ATB	0.000000000000
B-52CMC	*	B-52CMC	0.000135870987
B-52CMC	*	FB111PEN	-0.000003229885
B-52CMC	*	B1B	-0.000004926108
B-52CMC	*	ATB	0.000000347608
FB111PEN	*	FB111PEN	-0.001523388889
FB111PEN	*	B1B	-0.000000590476
FB111PEN	*	ATB	-0.000002349462

The decoded parameter estimates for Y2 are:

B1B	* B1B	-0.002660873469
B1B	* ATB	-0.001828917051
ATB	* ATB	-0.002460684183

The decoded parameter estimates for Y3 are:

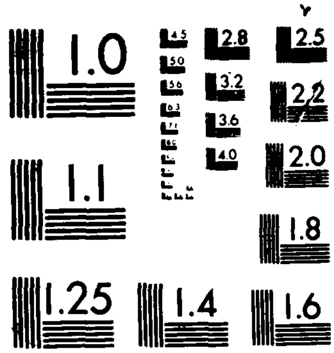
INTERCEPT		108.617774456922
MMII		-0.006888984802
MMIII		-0.034608905814
PKEEP		-0.188186919048
MICBM		-0.010141193473
POSC3		0.033040349702
POSTRIC4		-0.014357538219
TRID5		-0.042884634142
B-52PEN		-0.057963822172
B-52CMC		-0.230260844540
FB111PEN		-0.087682051040
B1B		-0.237347513019
ATB		-0.103181827207
MMII	* MMII	0.000005649778
MMII	* MMIII	-0.000004888889
MMII	* PKEEP	0.000001722667
MMII	* MICBM	0.000009055556
MMII	* POSC3	0.000003732639
MMII	* POSTRIC4	0.000012577160
MMII	* TRID5	0.000003546561
MMII	* B-52PEN	0.000007300855
MMII	* B-52CMC	-0.000004358621
MMII	* FB111PEN	-0.000005834074
MMII	* B1B	-0.000014364444
MMII	* ATB	0.000022401434
MMIII	* MMIII	0.000010788543
MMIII	* PKEEP	0.000128000000
MMIII	* MICBM	0.000014500000
MMIII	* POSC3	0.000004427083
MMIII	* POSTRIC4	0.000018595679
MMIII	* TRID5	0.000019874339
MMIII	* B-52PEN	0.000003525926
MMIII	* B-52CMC	0.000075670498
MMIII	* FB111PEN	-0.000015000000
MMIII	* B1B	0.000092857143
MMIII	* ATB	0.000010394265
PKEEP	* PKEEP	0.000328720000
PKEEP	* MICBM	-0.000032250000
PKEEP	* POSC3	0.000059181250
PKEEP	* POSTRIC4	0.000207638889
PKEEP	* TRID5	0.000133333333

The decoded parameter estimates for Y3 are:

PKEEP	*	B-52PEN	0.000175961538
PKEEP	*	B-52CMC	0.000226400000
PKEEP	*	FB111PEN	-0.000114586667
PKEEP	*	B1B	0.000036782857
PKEEP	*	ATB	0.000142138710
MICBM	*	MICBM	-0.000022273000
MICBM	*	POSC3	0.000035254687
MICBM	*	POSTRIC4	0.000002083333
MICBM	*	TRID5	-0.000002529762
MICBM	*	B-52PEN	0.000035335897
MICBM	*	B-52CMC	-0.000024137931
MICBM	*	FB111PEN	0.000070103333
MICBM	*	B1B	-0.000112857143
MICBM	*	ATB	0.000110282258
POSC3	*	POSC3	-0.000064106445
POSC3	*	POSTRIC4	-0.000100774740
POSC3	*	TRID5	-0.000008138021
POSC3	*	B-52PEN	0.000045947516
POSC3	*	B-52CMC	0.000163828125
POSC3	*	FB111PEN	0.000110348958
POSC3	*	B1B	0.000075892857
POSC3	*	ATB	-0.000022525202
POSTRIC4	*	POSTRIC4	0.000020420525
POSTRIC4	*	TRID5	0.000007905919
POSTRIC4	*	B-52PEN	0.000049857550
POSTRIC4	*	B-52CMC	0.000046546935
POSTRIC4	*	FB111PEN	-0.000006657407
POSTRIC4	*	B1B	0.000077130952
POSTRIC4	*	ATB	0.000037452957
TRID5	*	TRID5	0.000078050241
TRID5	*	B-52PEN	-0.000010921856
TRID5	*	B-52CMC	0.000183317939
TRID5	*	FB111PEN	0.000043527778
TRID5	*	B1B	0.000035819728
TRID5	*	ATB	-0.000037022849
B-52PEN	*	B-52PEN	0.000052172913
B-52PEN	*	B-52CMC	0.000071839080
B-52PEN	*	FB111PEN	0.000111914530
B-52PEN	*	B1B	0.000053802198
B-52PEN	*	ATB	0.000031017370
B-52CMC	*	B-52CMC	0.000178151011
B-52CMC	*	FB111PEN	0.000014011494
B-52CMC	*	B1B	0.00079279[illegible]
B-52CMC	*	ATB	0.000294424[illegible]60
FB111PEN	*	FB111PEN	0.0003886333[illegible]
FB111PEN	*	B1B	0.000035714[illegible]
FB111PEN	*	ATB	0.00010[illegible]
B1B	*	B1B	0.0006[illegible]
B1B	*	ATB	[illegible]
ATB	*	ATB	[illegible]

AD-A185 247 AUTOMATING RESPONSE SURFACE METHODOLOGY FOR THE ARSENAL 4/4
EXCHANGE MODEL(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. G E YIELDING

UNCLASSIFIED MAY 87 AFIT/GOR/ENS/86D-17 F/G 12/4 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

The decoded parameter estimates for Y4 are:

INTERCEPT			-358.700617186886
MMII			0.012140779900
MMIII			0.102736746640
PKEEP			0.749967267105
MICBM			0.157526907649
POSC3			0.783224201218
POSTRIC4			0.504962326655
TRID5			0.279168491762
B-52PEN			0.497093737632
B-52CMC			1.052439899331
FB111PEN			0.447496049808
B1B			0.965873900930
ATB			0.506347899584
MMII	*	MMII	0.000026359506
MMII	*	MMIII	0.000018641975
MMII	*	PKEEP	0.000023500444
MMII	*	MICBM	0.000016083556
MMII	*	POSC3	-0.000033940972
MMII	*	POSTRIC4	-0.000045022840
MMII	*	TRID5	-0.000005307407
MMII	*	B-52PEN	-0.000006267806
MMII	*	B-52CMC	0.000000048276
MMII	*	FB111PEN	0.000000462222
MMII	*	B1B	-0.000002063492
MMII	*	ATB	-0.000078628674
MMIII	*	MMIII	0.000041107951
MMIII	*	PKEEP	-0.000088055111
MMIII	*	MICBM	0.000043444444
MMIII	*	POSC3	-0.000077256944
MMIII	*	POSTRIC4	-0.000076582099
MMIII	*	TRID5	-0.000024371693
MMIII	*	B-52PEN	-0.000058083761
MMIII	*	B-52CMC	-0.000203016858
MMIII	*	FB111PEN	-0.000067777778
MMIII	*	B1B	-0.000477301587
MMIII	*	ATB	-0.000162141935
PKEEP	*	PKEEP	0.000867312000
PKEEP	*	MICBM	0.000017126000
PKEEP	*	POSC3	-0.000884375000
PKEEP	*	POSTRIC4	-0.000786630556
PKEEP	*	TRID5	0.000211086905
PKEEP	*	B-52PEN	-0.000545033333
PKEEP	*	B-52CMC	-0.001017024138
PKEEP	*	FB111PEN	-0.000532920000
PKEEP	*	B1B	-0.000976428571
PKEEP	*	ATB	-0.000870564516
MICBM	*	MICBM	-0.000092820000
MICBM	*	POSC3	-0.000270604687
MICBM	*	POSTRIC4	-0.000131683333
MICBM	*	TRID5	0.000032031548

The decoded parameter estimates for Y4 are:

MICBM	*	B-52PEN	-0.000142548718
MICBM	*	B-52CMC	0.000013468966
MICBM	*	FB111PEN	-0.000168436667
MICBM	*	B1B	-0.000049642857
MICBM	*	ATB	-0.000261391935
POSC3	*	POSC3	-0.000378417969
POSC3	*	POSTRIC4	-0.000401881510
POSC3	*	TRID5	-0.000330054874
POSC3	*	B-52PEN	-0.000518954327
POSC3	*	B-52CMC	-0.000772998384
POSC3	*	FB111PEN	-0.000597333333
POSC3	*	B1B	-0.000618723214
POSC3	*	ATB	-0.000346207157
POSTRIC4	*	POSTRIC4	-0.000057056327
POSTRIC4	*	TRID5	-0.000191333499
POSTRIC4	*	B-52PEN	-0.000330416667
POSTRIC4	*	B-52CMC	-0.000532957375
POSTRIC4	*	FB111PEN	-0.000100407407
POSTRIC4	*	B1B	-0.000732888889
POSTRIC4	*	ATB	-0.000586917563
TRID5	*	TRID5	0.000156053359
TRID5	*	B-52PEN	-0.000172752595
TRID5	*	B-52CMC	-0.000690874384
TRID5	*	FB111PEN	-0.000179686508
TRID5	*	B1B	-0.000594812925
TRID5	*	ATB	-0.000256035906
B-52PEN	*	B-52PEN	-0.000131569691
B-52PEN	*	B-52CMC	-0.000379778957
B-52PEN	*	FB111PEN	-0.000236914530
B-52PEN	*	B1B	-0.000504120879
B-52PEN	*	ATB	-0.000352694376
B-52CMC	*	B-52CMC	-0.000628391795
B-52CMC	*	FB111PEN	-0.000221620690
B-52CMC	*	B1B	-0.001846985222
B-52CMC	*	ATB	-0.000938020022
FB111PEN	*	FB111PEN	0.000322833333
FB111PEN	*	B1B	-0.001125000000
FB111PEN	*	ATB	-0.000373317204
B1B	*	B1B	0.000874742857
B1B	*	ATB	0.000195852535
ATB	*	ATB	0.001210369407

The decoded parameter estimates for Y5 are:

INTERCEPT	-125.214476892050
MMII	0.012928076306
MMIII	0.054679718141

The decoded parameter estimates for Y5 are:

PKEEP		0.387007125205
MICBM		0.069343867570
POSC3		0.333503145141
POSTRIC4		0.233209622389
TRID5		0.185094193969
B-52PEN		0.189473105676
B-52CMC		0.380679221401
FB111PEN		0.183295167488
B1B		0.469928564274
ATB		0.318958846779
MMII	* MMII	0.000004993778
MMII	* MMIII	0.000001975309
MMII	* PKEEP	-0.000010111111
MMII	* MICBM	0.000001638667
MMII	* POSC3	-0.000014496528
MMII	* POSTRIC4	-0.000016242593
MMII	* TRID5	-0.000009449471
MMII	* B-52PEN	-0.000001531624
MMII	* B-52CMC	-0.000001149425
MMII	* FB111PEN	-0.000000925926
MMII	* B1B	-0.000003729524
MMII	* ATB	-0.000026030108
MMIII	* MMIII	0.000008793284
MMIII	* PKEEP	-0.000052610667
MMIII	* MICBM	0.000005861333
MMIII	* POSC3	-0.000030729167
MMIII	* POSTRIC4	-0.000030594444
MMIII	* TRID5	-0.000029199735
MMIII	* B-52PEN	-0.000023486610
MMIII	* B-52CMC	-0.000069348659
MMIII	* FB111PEN	-0.000031204444
MMIII	* B1B	-0.000175634286
MMIII	* ATB	-0.000068458781
PKEEP	* PKEEP	0.000104908000
PKEEP	* MICBM	-0.000004500000
PKEEP	* POSC3	-0.000359959375
PKEEP	* POSTRIC4	-0.000291144444
PKEEP	* TRID5	-0.000064583333
PKEEP	* B-52PEN	-0.000182212821
PKEEP	* B-52CMC	-0.000373920690
PKEEP	* FB111PEN	-0.000241666667
PKEEP	* B1B	-0.000408217143
PKEEP	* ATB	-0.000325200000
MICBM	* MICBM	-0.000011016000
MICBM	* POSC3	-0.000104003125
MICBM	* POSTRIC4	-0.000057031944
MICBM	* TRID5	-0.000008184524
MICBM	* B-52PEN	-0.000052484615
MICBM	* B-52CMC	-0.000000431034
MICBM	* FB111PEN	-0.000054373333

The decoded parameter estimates for Y5 are:

MICBM	*	B1B	-0.000051251429
MICBM	*	ATB	-0.000083972581
POSC3	*	POSC3	-0.000163688965
POSC3	*	POSTRIC4	-0.000197211372
POSC3	*	TRID5	-0.000143694196
POSC3	*	B-52PEN	-0.000192682292
POSC3	*	B-52CMC	-0.000279498922
POSC3	*	FB111PEN	-0.000219729167
POSC3	*	B1B	-0.000242325893
POSC3	*	ATB	-0.000160504032
POSTRIC4	*	POSTRIC4	-0.000055655864
POSTRIC4	*	TRID5	-0.000081018519
POSTRIC4	*	B-52PEN	-0.000122751068
POSTRIC4	*	B-52CMC	-0.000208034004
POSTRIC4	*	FB111PEN	-0.000031828704
POSTRIC4	*	B1B	-0.000276289683
POSTRIC4	*	ATB	-0.000232484319
TRID5	*	TRID5	-0.000022188563
TRID5	*	B-52PEN	-0.000073211996
TRID5	*	B-52CMC	-0.000248871100
TRID5	*	FB111PEN	-0.000067460317
TRID5	*	B1B	-0.000243942177
TRID5	*	ATB	-0.000116527458
B-52PEN	*	B-52PEN	-0.000054368836
B-52PEN	*	B-52CMC	-0.000139257294
B-52PEN	*	FB111PEN	-0.000075051282
B-52PEN	*	B1B	-0.000182919414
B-52PEN	*	ATB	-0.000136993383
B-52CMC	*	B-52CMC	-0.000188275862
B-52CMC	*	FB111PEN	-0.000096982759
B-52CMC	*	B1B	-0.000592364532
B-52CMC	*	ATB	-0.000338223026
FB111PEN	*	FB111PEN	-0.000099133333
FB111PEN	*	B1B	-0.000397028571
FB111PEN	*	ATB	-0.000133064516
B1B	*	B1B	-0.000055232653
B1B	*	ATB	-0.000243953917
ATB	*	ATB	0.000039081686

Appendix U: Postulated Model Completion For The Fifth Objective

The decoded parameter estimates are "fitted" with their parameters to make up the completed postulated model for the fifth objective. The parameter estimates are taken directly from the decoded parameter estimates listed for the fifth objective in Appendix T.

```
Y5 = - 125.214476892050
     + 0.012928076306 * (MMII)
     + 0.054679718141 * (MMIII)
     + 0.387007125205 * (PKEEP)
     + 0.069343867570 * (MICBM)
     + 0.333503145141 * (POSC3)
     + 0.233209622389 * (POSTRIC4)
     + 0.185094193969 * (TRID5)
     + 0.189473105676 * (B52PEN)
     + 0.380679221401 * (B52CMC)
     + 0.183295167488 * (FB111PEN)
     + 0.469928564274 * (B1B)
     + 0.318958846779 * (ATB)
     + 0.000004993778 * (MMII) * (MMII)
     + 0.000001975309 * (MMII) * (MMIII)
     - 0.000010111111 * (MMII) * (PKEEP)
     + 0.000001638667 * (MMII) * (MICBM)
     - 0.000014496528 * (MMII) * (POSC3)
     - 0.000016242593 * (MMII) * (POSTRIC4)
     - 0.000009449471 * (MMII) * (TRID5)
     - 0.000001531624 * (MMII) * (B52PEN)
     - 0.000001149425 * (MMII) * (B52CMC)
     - 0.000000925926 * (MMII) * (FB111PEN)
     - 0.000003729524 * (MMII) * (B1B)
     - 0.000026030108 * (MMII) * (ATB)
     + 0.000008793284 * (MMIII) * (MMIII)
     - 0.000052610667 * (MMIII) * (PKEEP)
     + 0.000005861333 * (MMIII) * (MICBM)
     - 0.000030729167 * (MMIII) * (POSC3)
     - 0.000030594444 * (MMIII) * (POSTRIC4)
     - 0.000029199735 * (MMIII) * (TRID5)
     - 0.000023486610 * (MMIII) * (B52PEN)
     - 0.000069348659 * (MMIII) * (B52CMC)
     - 0.000031204444 * (MMIII) * (FB111PEN)
     - 0.000175634286 * (MMIII) * (B1B)
     - 0.000068458781 * (MMIII) * (ATB)
     + 0.000104908000 * (PKEEP) * (PKEEP)
     - 0.000004500000 * (PKEEP) * (MICBM)
```

- 0.000359959375 * (PKEEP) * (POSC3)
- 0.000291144444 * (PKEEP) * (POSTRIC4)
- 0.000064583333 * (PKEEP) * (TRID5)
- 0.000182212821 * (PKEEP) * (B52PEN)
- 0.000373920690 * (PKEEP) * (B52CMC)
- 0.000241666667 * (PKEEP) * (FB111PEN)
- 0.000408217143 * (PKEEP) * (B1B)
- 0.000325200000 * (PKEEP) * (ATB)
- 0.000011016000 * (MICBM) * (MICBM)
- 0.000104003125 * (MICBM) * (POSC3)
- 0.000057031944 * (MICBM) * (POSTRIC4)
- 0.000008184524 * (MICBM) * (TRID5)
- 0.000052484615 * (MICBM) * (B52PEN)
- 0.000000431034 * (MICBM) * (B52CMC)
- 0.000054373333 * (MICBM) * (FB111PEN)
- 0.000051251429 * (MICBM) * (B1B)
- 0.000083972581 * (MICBM) * (ATB)
- 0.000163688965 * (POSC3) * (POSC3)
- 0.000197211372 * (POSC3) * (POSTRIC4)
- 0.000143694196 * (POSC3) * (TRID5)
- 0.000192682292 * (POSC3) * (B52PEN)
- 0.000279498922 * (POSC3) * (B52CMC)
- 0.000219729167 * (POSC3) * (FB111PEN)
- 0.000242325893 * (POSC3) * (B1B)
- 0.000160504032 * (POSC3) * (ATB)
- 0.000055655864 * (POSTRIC4) * (POSTRIC4)
- 0.000081018519 * (POSTRIC4) * (TRID5)
- 0.000122751068 * (POSTRIC4) * (B52PEN)
- 0.000208034004 * (POSTRIC4) * (B52CMC)
- 0.000031828704 * (POSTRIC4) * (FB111PEN)
- 0.000276289683 * (POSTRIC4) * (B1B)
- 0.000232484319 * (POSTRIC4) * (ATB)
- 0.000022188563 * (TRID5) * (TRID5)
- 0.000073211996 * (TRID5) * (B52PEN)
- 0.000248871100 * (TRID5) * (B52CMC)
- 0.000067460317 * (TRID5) * (FB111PEN)
- 0.000243942177 * (TRID5) * (B1B)
- 0.000116527458 * (TRID5) * (ATB)
- 0.000054368836 * (B52PEN) * (B52PEN)
- 0.000139257294 * (B52PEN) * (B52CMC)
- 0.000075051282 * (B52PEN) * (FB111PEN)
- 0.000182919414 * (B52PEN) * (B1B)
- 0.000136993383 * (B52PEN) * (ATB)
- 0.000188275862 * (B52CMC) * (B52CMC)
- 0.000096982759 * (B52CMC) * (FB111PEN)
- 0.000592364532 * (B52CMC) * (B1B)
- 0.000338223026 * (B52CMC) * (ATB)
- 0.000099133333 * (FB111PEN) * (FB111PEN)
- 0.000397028571 * (FB111PEN) * (B1B)
- 0.000133064516 * (FB111PEN) * (ATB)
- 0.000055232653 * (B1B) * (B1B)
- 0.000243953917 * (B1B) * (ATB)
+ 0.000039081686 * (ATB) * (ATB)

Appendix V: Regression Output For The Reduced Postulated Model

The full postulated model for the fifth objective (regression output is shown in Appendix O) is reduced from 90 variables to 16 variables (INTERCEPT not included). The regression output for the reduced model is shown below. The deleted variables are shown contributing nothing (zero) to the model sum-of-squares (SSR). The reduced variables are shown last. Compare the results of this appendix with the results for the full model in Appendix O.

Regression Output for Dependent Variable: Y5

Model Sum of Squares (SSR)	Error Sum of Squares (SSE)	Total Sum of Squares (SSTO)	R-Square
10274.32	780.20	11054.52	0.9294

Parameters Of Y5 Sorted In Ascending Order
By Their Sum Of Squares

Number	Variable	Type I Sum of Squares	Percent Of SSR	Summed Percent Of SSR
1	MMII * MMII	0.000	0.00	0.00
2	MMIII * MMIII	0.000	0.00	0.00
3	PKEEP * PKEEP	0.000	0.00	0.00
4	MICBM * MICBM	0.000	0.00	0.00
5	POSC3 * POSC3	0.000	0.00	0.00
6	POSTRIC4 * POSTRIC4	0.000	0.00	0.00
7	TRID5 * TRID5	0.000	0.00	0.00
8	B-52PEN * B-52PEN	0.000	0.00	0.00
9	B-52CMC * B-52CMC	0.000	0.00	0.00
10	FB111PEN * FB111PEN	0.000	0.00	0.00
11	B1B * B1B	0.000	0.00	0.00
12	ATB * ATB	0.000	0.00	0.00
13	MMII * MMIII	0.000	0.00	0.00
14	MMII * PKEEP	0.000	0.00	0.00
15	MMII * MICBM	0.000	0.00	0.00

Parameters Of Y5 Sorted In Ascending Order By Their Sum Of Squares

Number	Variable		Type I Sum of Squares	Percent Of SSR	Summed Percent Of SSR
16	MMII	* POSC3	0.000	0.00	0.00
17	MMII	* POSTRIC4	0.000	0.00	0.00
18	MMII	* TRID5	0.000	0.00	0.00
19	MMII	* B-52PEN	0.000	0.00	0.00
20	MMII	* B-52CMC	0.000	0.00	0.00
21	MMII	* FB111PEN	0.000	0.00	0.00
22	MMII	* B1B	0.000	0.00	0.00
23	MMII	* ATB	0.000	0.00	0.00
24	MMIII	* PKEEP	0.000	0.00	0.00
25	MMIII	* MICBM	0.000	0.00	0.00
26	MMIII	* POSC3	0.000	0.00	0.00
27	MMIII	* POSTRIC4	0.000	0.00	0.00
28	MMIII	* TRID5	0.000	0.00	0.00
29	MMIII	* B-52PEN	0.000	0.00	0.00
30	MMIII	* B-52CMC	0.000	0.00	0.00
31	MMIII	* FB111PEN	0.000	0.00	0.00
32	MMIII	* B1B	0.000	0.00	0.00
33	MMIII	* ATB	0.000	0.00	0.00
34	PKEEP	* MICBM	0.000	0.00	0.00
35	PKEEP	* POSC3	0.000	0.00	0.00
36	PKEEP	* POSTRIC4	0.000	0.00	0.00
37	PKEEP	* TRID5	0.000	0.00	0.00
38	PKEEP	* B-52PEN	0.000	0.00	0.00
39	PKEEP	* FB111PEN	0.000	0.00	0.00
40	PKEEP	* B1B	0.000	0.00	0.00
41	PKEEP	* ATB	0.000	0.00	0.00
42	MICBM	* POSC3	0.000	0.00	0.00
43	MICBM	* POSTRIC4	0.000	0.00	0.00
44	MICBM	* TRID5	0.000	0.00	0.00
45	MICBM	* B-52PEN	0.000	0.00	0.00
46	MICBM	* B-52CMC	0.000	0.00	0.00
47	MICBM	* FB111PEN	0.000	0.00	0.00
48	MICBM	* B1B	0.000	0.00	0.00
49	MICBM	* ATB	0.000	0.00	0.00
50	POSC3	* POSTRIC4	0.000	0.00	0.00
51	POSC3	* B-52PEN	0.000	0.00	0.00
52	POSC3	* B-52CMC	0.000	0.00	0.00
53	POSC3	* FB111PEN	0.000	0.00	0.00
54	POSC3	* B1B	0.000	0.00	0.00
55	POSC3	* ATB	0.000	0.00	0.00
56	POSTRIC4	* TRID5	0.000	0.00	0.00
57	POSTRIC4	* B-52PEN	0.000	0.00	0.00
58	POSTRIC4	* B-52CMC	0.000	0.00	0.00
59	POSTRIC4	* FB111PEN	0.000	0.00	0.00
60	POSTRIC4	* B1B	0.000	0.00	0.00
61	TRID5	* B-52PEN	0.000	0.00	0.00

Parameters Of Y5 Sorted In Ascending Order
By Their Sum Of Squares

Number	Variable	Type I Sum of Squares	Percent Of SSR	Summed Percent Of SSR
62	TRID5 * FB111PEN	0.000	0.00	0.00
63	TRID5 * B1B	0.000	0.00	0.00
64	TRID5 * ATB	0.000	0.00	0.00
65	B-52PEN * B-52CMC	0.000	0.00	0.00
66	B-52PEN * FB111PEN	0.000	0.00	0.00
67	B-52PEN * B1B	0.000	0.00	0.00
68	B-52PEN * ATB	0.000	0.00	0.00
69	B-52CMC * FB111PEN	0.000	0.00	0.00
70	B-52CMC * B1B	0.000	0.00	0.00
71	B-52CMC * ATB	0.000	0.00	0.00
72	FB111PEN * B1B	0.000	0.00	0.00
73	FB111PEN * ATB	0.000	0.00	0.00
74	B1B * ATB	0.000	0.00	0.00
75	MMII	33.698	0.33	0.33
76	POSTRIC4 * ATB	34.466	0.34	0.66
77	PKEEP * B-52CMC	37.628	0.37	1.03
78	POSC3 * TRID5	38.192	0.37	1.40
79	FB111PEN	39.125	0.38	1.78
80	MICBM	74.025	0.72	2.50
81	TRID5 * B-52CMC	94.090	0.92	3.42
82	B-52PEN	253.805	2.47	5.89
83	MMIII	446.900	4.35	10.24
84	POSTRIC4	541.260	5.27	15.51
85	B-52CMC	656.000	6.38	21.89
86	B1B	701.521	6.83	28.72
87	POSC3	733.597	7.14	35.86
88	PKEEP	1150.397	11.20	47.06
89	ATB	1448.373	14.10	61.15
90	TRID5	3991.239	38.85	100.00

For Y5: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
1	85.75	87.78	-2.03
2	74.36	75.08	-0.72
3	68.57	65.21	3.36
4	82.90	84.10	-1.20
5	80.32	78.81	1.51
6	83.35	82.50	0.85
7	82.53	81.04	1.49
8	85.01	86.33	-1.32
9	73.23	73.62	-0.39

For Y5: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
10	64.12	63.76	0.36
11	82.16	82.65	-0.49
12	79.48	77.36	2.12
13	56.77	58.48	-1.71
14	67.57	68.34	-0.77
15	69.47	69.79	-0.32
16	58.67	59.93	-1.26
17	83.88	86.46	-2.58
18	82.44	82.47	-0.03
19	79.43	76.66	2.77
20	81.03	80.64	0.39
21	73.86	72.16	1.70
22	78.27	77.98	0.29
23	74.90	72.69	2.21
24	81.25	81.17	0.08
25	79.58	77.19	2.39
26	72.99	71.37	1.62
27	78.11	75.35	2.76
28	66.84	66.88	-0.04
29	59.61	62.89	-3.28
30	69.76	68.71	1.05
31	76.52	73.99	2.53
32	66.78	68.18	-1.40
33	86.08	87.58	-1.50
34	84.25	88.19	-3.94
35	63.71	67.55	-3.84
36	75.38	76.63	-1.25
37	74.12	74.48	-0.36
38	85.41	85.43	-0.02
39	81.10	79.12	1.98
40	81.88	81.27	0.61
41	79.79	77.55	2.24
42	57.17	56.90	0.27
43	68.83	70.32	-1.49
44	67.53	68.17	-0.64
45	55.86	54.75	1.11
46	78.97	75.40	3.57
47	83.59	86.04	-2.45
48	62.45	65.40	-2.95
49	80.80	81.91	-1.11
50	80.16	80.34	-0.18
51	78.33	76.36	1.97
52	79.09	77.93	1.16
53	70.63	71.15	-0.52
54	77.41	75.14	2.27
55	72.33	72.99	-0.66
56	79.60	79.76	-0.16
57	78.84	78.19	0.65

For Y5: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
58	72.92	74.21	-1.29
59	75.95	75.77	0.18
60	65.07	69.00	-3.93
61	61.98	67.44	-5.46
62	69.23	71.42	-2.19
63	74.75	73.57	1.18
64	67.54	69.59	-2.05
65	83.37	87.48	-4.11
66	79.47	80.86	-1.39
67	75.37	74.46	0.91
68	81.13	81.08	0.05
69	76.76	75.26	1.50
70	80.34	81.66	-1.32
71	75.53	74.89	0.64
72	80.34	80.71	-0.37
73	76.29	74.09	2.20
74	64.53	67.68	-3.15
75	76.02	74.31	1.71
76	65.92	68.49	-2.57
77	54.30	61.87	-7.57
78	66.28	68.27	-1.99
79	76.39	75.04	1.35
80	65.24	68.64	-3.40
81	86.48	89.98	-3.50
82	79.88	82.54	-2.66
83	79.38	80.98	-1.60
84	85.98	88.42	-2.44
85	74.40	72.62	1.78
86	75.00	74.19	0.81
87	71.84	70.45	1.39
88	83.84	86.24	-2.40
89	77.09	74.65	2.44
90	76.55	73.09	3.46
91	83.35	84.68	-1.33
92	70.66	68.88	1.78
93	54.97	57.29	-2.32
94	58.06	58.86	-0.80
95	67.65	66.75	0.90
96	65.05	65.18	-0.13
97	85.18	88.60	-3.42
98	81.68	81.98	-0.30
99	80.20	77.99	2.21
100	83.76	84.62	-0.86
101	70.65	68.82	1.83
102	73.83	72.81	1.02
103	72.04	71.35	0.69
104	84.45	87.15	-2.70
105	80.93	80.53	0.40

For Y5: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
106	79.38	76.54	2.84
107	83.03	83.17	-0.14
108	68.51	67.37	1.14
109	57.26	60.75	-3.49
110	64.52	64.73	-0.21
111	66.66	66.18	0.48
112	59.41	62.20	-2.79
113	83.45	87.27	-3.82
114	76.56	77.75	-1.19
115	74.34	71.35	2.99
116	81.50	80.86	0.64
117	79.87	76.88	2.99
118	81.96	83.28	-1.32
119	78.96	78.00	0.96
120	80.57	81.98	-1.41
121	73.43	72.47	0.96
122	64.53	66.06	-1.53
123	78.45	75.58	2.87
124	74.62	71.60	3.02
125	57.24	62.08	-4.84
126	69.35	68.48	0.87
127	74.92	73.77	1.15
128	68.19	67.37	0.82
129	82.11	82.54	-0.43
130	81.54	80.97	0.57
131	79.43	76.74	2.69
132	80.01	78.30	1.71
133	66.49	67.66	-1.17
134	76.30	76.23	0.07
135	75.18	74.78	0.40
136	81.28	81.09	0.19
137	80.69	79.52	1.17
138	78.53	75.29	3.24
139	79.14	76.85	2.29
140	65.00	66.20	-1.20
141	62.43	64.64	-2.21
142	73.65	73.21	0.44
143	75.09	74.66	0.43
144	63.65	66.09	-2.44
145	81.49	82.48	-0.99
146	77.65	75.86	1.79
147	77.04	74.30	2.74
148	80.91	80.92	-0.01
149	79.96	78.77	1.19
150	80.54	80.33	0.21
151	77.37	75.05	2.32
152	78.50	77.20	1.30
153	69.46	70.58	-1.12

For Y5: Residuals Associated With The Design Points

Design Point	AEM's Response	Postulated Model's Response	Residual
154	66.37	69.01	-2.64
155	77.52	75.64	1.88
156	76.39	73.49	2.90
157	65.23	66.86	-1.63
158	68.32	68.43	-0.11
159	76.47	73.71	2.76
160	73.47	72.15	1.32
161	85.51	90.37	-4.86
162	79.69	80.85	-1.16
163	75.20	74.23	0.97
164	82.94	83.75	-0.81
165	79.89	76.97	2.92
166	82.66	83.60	-0.94
167	77.07	75.12	1.95
168	80.15	81.89	-1.74
169	73.55	72.37	1.18
170	66.68	65.75	0.93
171	76.99	75.27	1.72
172	69.67	68.50	1.17
173	55.80	58.98	-3.18
174	64.22	65.60	-1.38
175	76.60	74.08	2.52
176	72.01	67.46	4.55
177	82.68	83.10	-0.42
178	75.67	75.66	0.01
179	70.73	67.77	2.96
180	79.73	79.36	0.37
181	78.68	77.21	1.47
182	81.82	80.95	0.87
183	80.75	79.50	1.25
184	81.84	81.65	0.19
185	74.58	74.21	0.37
186	66.38	66.32	0.06
187	78.74	77.91	0.83
188	77.51	75.76	1.75
189	62.57	64.17	-1.60
190	72.61	72.06	0.55
191	74.48	73.51	0.97
192	65.14	65.62	-0.48
193	75.24	74.67	0.57

Appendix V: Decoded Coefficients For The Reduced Postulated Model

Taken from the DECODE.OUT file, the decoded coefficients for the fifth objective's reduced postulated model are shown below. Compare this output with the decoded output for the full postulated model for the fifth objective shown in Appendix T.

The decoded parameter estimates for Y5 are:

INTERCEPT		-3.687629292556
MMII		0.003225022222
MMIII		0.011744444444
PKEEP		0.121438027586
MICBM		0.010754700000
POSC3		0.077041093750
POSTRIC4		0.054804722222
TRID5		0.105882105911
B-52PEN		0.025530897436
B-52CMC		0.115705689655
FB111PEN		0.026062666667
B1B		0.094593714286
ATB		0.149263978495
MMII	* MMII	0.000000000000
MMII	* MMIII	0.000000000000
MMII	* PKEEP	0.000000000000
MMII	* MICBM	0.000000000000
MMII	* POSC3	0.000000000000
MMII	* POSTRIC4	0.000000000000
MMII	* TRID5	0.000000000000
MMII	* B-52PEN	0.000000000000
MMII	* B-52CMC	0.000000000000
MMII	* FB111PEN	0.000000000000
MMII	* B1B	0.000000000000
MMII	* ATB	0.000000000000
MMIII	* MMIII	0.000000000000
MMIII	* PKEEP	0.000000000000
MMIII	* MICBM	0.000000000000
MMIII	* POSC3	0.000000000000
MMIII	* POSTRIC4	0.000000000000
MMIII	* TRID5	0.000000000000
MMIII	* B-52PEN	0.000000000000
MMIII	* B-52CMC	0.000000000000
MMIII	* FB111PEN	0.000000000000
MMIII	* B1B	0.000000000000
MMIII	* ATB	0.000000000000
PKEEP	* PKEEP	0.000000000000
PKEEP	* MICBM	0.000000000000

PKEEP	*	POSC3	0.000000000000
PKEEP	*	POSTRIC4	0.000000000000
PKEEP	*	TRID5	0.000000000000
PKEEP	*	B-52PEN	0.000000000000
PKEEP	*	B-52CMC	-0.000373920690
PKEEP	*	FB111PEN	0.000000000000
PKEEP	*	B1B	0.000000000000
PKEEP	*	ATB	0.000000000000
MICBM	*	MICBM	0.000000000000
MICBM	*	POSC3	0.000000000000
MICBM	*	POSTRIC4	0.000000000000
MICBM	*	TRID5	0.000000000000
MICBM	*	B-52PEN	0.000000000000
MICBM	*	B-52CMC	0.000000000000
MICBM	*	FB111PEN	0.000000000000
MICBM	*	B1B	0.000000000000
MICBM	*	ATB	0.000000000000
POSC3	*	POSC3	0.000000000000
POSC3	*	POSTRIC4	0.000000000000
POSC3	*	TRID5	-0.000143694196
POSC3	*	B-52PEN	0.000000000000
POSC3	*	B-52CMC	0.000000000000
POSC3	*	FB111PEN	0.000000000000
POSC3	*	B1B	0.000000000000
POSC3	*	ATB	0.000000000000
POSTRIC4	*	POSTRIC4	0.000000000000
POSTRIC4	*	TRID5	0.000000000000
POSTRIC4	*	B-52PEN	0.000000000000
POSTRIC4	*	B-52CMC	0.000000000000
POSTRIC4	*	FB111PEN	0.000000000000
POSTRIC4	*	B1B	0.000000000000
POSTRIC4	*	ATB	-0.000232484319
TRID5	*	TRID5	0.000000000000
TRID5	*	B-52PEN	0.000000000000
TRID5	*	B-52CMC	-0.000248871100
TRID5	*	FB111PEN	0.000000000000
TRID5	*	B1B	0.000000000000
TRID5	*	ATB	0.000000000000
B-52PEN	*	B-52PEN	0.000000000000
B-52PEN	*	B-52CMC	0.000000000000
B-52PEN	*	FB111PEN	0.000000000000
B-52PEN	*	B1B	0.000000000000
B-52PEN	*	ATB	0.000000000000
B-52CMC	*	B-52CMC	0.000000000000
B-52CMC	*	FB111PEN	0.000000000000
B-52CMC	*	B1B	0.000000000000
B-52CMC	*	ATB	0.000000000000
FB111PEN	*	FB111PEN	0.000000000000
FB111PEN	*	B1B	0.000000000000
FB111PEN	*	ATB	0.000000000000
B1B	*	B1B	0.000000000000
B1B	*	ATB	0.000000000000
ATB	*	ATB	0.000000000000

Bibliography

1. Box, G. E. P. and D. W. Behnken. "Some New Three Level Designs for the Study of Quantitative Variables," Technometrics, 2: 455-475 (November 1960).

2. Box, G. E. P. and N. R. Draper. "A Basis For The Selection Of A Response Surface Design," Journal Of The American Statistical Association, 54: 622-654 (September 1959).

3. Box, G. E. P. and N. R. Draper. Empirical Model-Building and Response Surfaces. New York: John Wiley & Sons, 1987.

4. Cooke, James C. Optimizing Force Structure And Force Deployment For The Rapid Deployment Force. MS Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, March 1984 (AD-A141148).

5. Cotsworth, W. L. The Arsenal Exchange Model (AEM) Management Summary. Software Documentation. HQ USAF/SA, The Pentagon, Washington D. C., 31 July 1983.

6. Cotsworth, W. L. et al. The Arsenal Exchange Model (AEM) User's Manual. User's Manual. HQ USAF/SA, The Pentagon, Washington D. C., June 1985.

7. Davies, Owen L. The Design And Analysis Of Industrial Experiments. New York: Hafner Publishing Company, 1963.

8. Department of the Air Force. US Air Force Glossary of Standardized Terms. AFM 11-1, Vol I. Washington: HQ USAF, 2 January 1976.

9. Donovan, James L. An Application Of Response Surface Methodology To A Macroeconomic Model. MS Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1985.

10. Draper, Norman and Harry Smith. Applied Regression Analysis. New York: John Wiley & Son, 1981.

11. Graney, Robert E. An Optimization Methodology For Multi-Criteria Comparison Using Response Surface Methods And Mathematical Programming. MS Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, March 1984 (AD-A141169).

12. Hillier, Frederick S. and Gerald J. Lieberman. Introduction to Operations Research. Oakland, CA: Holden-Day, Inc., 1980.

13. Ignizio, James P. Linear Programming In Single- & Multiple-Objective Systems. Englewood Cliffs, NJ: Prentice Hall, Inc., 1982.

14. Ishihara, Bryan K. Evaluating Experimental Designs For Fitting Response Surfaces Of Deterministic Models. MS Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1985.

15. Johnson, William C. ALLOT: An Enhancement. MS Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, September 1985.

16. Leonard, Maj. Telephone Conversation. AF/XOXR-MAA, Washington D.C., 3 July 86.

17. Libby, Charles W. An Analysis Of Strategic Nuclear Force Structures And A Policy For Launch Before Impact For US ICBMs. MS Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, May 1985 (AD-B095201).

18. Manacapilli, Thomas W. A Methodology For Identifying Cost Effective Strategic Force Mixes. MS Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1984 (AD-A151773).

19. Meitzler, Thomas D. An Application Of The Response Surface Methodology To A Readiness-Analysis Problem. MS Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1984 (AD-B090391).

20. Montgomery, Douglas C. Design And Analysis Of Experiments. New York: John Wiley & Sons, 1976.

21. Myers, Raymond H. Response Surface Methodology. Virginia Polytechnic Institute and State University, 1976.

22. Neter, John et al. Applied Linear Statistical Models. Homewood, IL: Richard D. Irwin, Inc., 1985.

23. Rowell, William F. An Improved Methodology For Air Force Wide Mission Area Analysis Evaluation Of U.S. Strategic Offensive Capability. Technical Paper to Air Force Wide Mission Area Analysis. Air Force Studies and Analysis, Washington DC, December 1981.

24. Shannon, Terry C. Introduction to VAX/VMS. Springhouse, Pennsylvania: Professional Press, Inc., 1985.

25. Smith, Palmer W. "A Methodology For Developing And Analyzing The Optimal Response Of An Economic Criteria To Simultaneously Induced Events Using Input-Output, Mathematical Programming, And Response Surface Methodologies," Unpublished PhD Dissertation, University of Alabama, July 1975.

26. Smith, Palmer W. and Mellichamp, Joseph M. "Multidimensional Parametric Analysis Using Response Surface Methodology And Mathematical Programming As Applied To Military Problems." Proceedings of the Pacific Conference on Operations Research, April 23-28, 1979, Seoul, Korea, Vol. I, pp. 592-615.

27. Smith, Palmer W. and Joseph M. Mellichamp. "A Methodology For Multidimensional Impact Analysis For Military Problems," Proceedings of the 53rd Military Operations Research Society. ppgs 55-63, 1985.

28. Sparrow, Kalla J. An Interactive Computer Package For Use With Simulation Models Which Performs Multidimensional Sensitivity Analysis By Employing The Techniques Of Response Surface Methodology. MS Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1984 (151956).

29. Steinberg, David M. and William G. Hunter. "Experimental Design: Review and Comment," Technometrics, 26: 71-97 (May 1984).

30. Wu, Nesa and Richard Coppins. Linear Programming And Extensions. New York: McGraw-Hill Book Company, 1981.

END

11-87

DTIC